



# POLITECNICO DI BARI

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE  
(DEI) CORSO DI LAUREA MAGISTRALE IN INGEGNERIA  
DELL'AUTOMAZIONE

---

## **Robotics 1<sup>st</sup> Module – Industrial handling**

“Motion Controls for a 2DOF planar manipulator”

### **Professor:**

Prof. Ing. Paolo Lino

### **Students:**

Savino Francesco

Savino Tommaso

# Sommario

|   |    |
|---|----|
| 1. INTRODUZIONE .....   | 5  |
| 2. MODELLO DINAMICO DEL MANIPOLATORE .....                                    | 6  |
| 2.1 Equazioni del moto del manipolatore.....                                  | 11 |
| 2.2 Implementazione Matlab e Simulink.....                                    | 13 |
| 2.3 Valori costanti per simulink .....  | 15 |
| 3. RIFERIMENTO E TRAIETTORIA .....  | 16 |
| 4. SINGOLARITÀ .....  | 19 |
| 5. CONTROLLO NELLO SPAZIO DEI GIUNTI.....                                     | 20 |
| 5.1 Controllo Decentralizzato.....  | 20 |
| 5.1.1 Implementazione controllo decentralizzato .....                         | 22 |
| 5.1.2 Implementazione Feedback control .....                                  | 26 |
| 5.1.3 Compensazione in avanti decentralizzata.....                            | 32 |
| 5.1.4 Compensazione in avanti a coppia precalcolata .....                     | 40 |
| 5.2 Controllo Centralizzato .....   | 49 |
| 5.2.1 Controllo PD con compensazione di gravità nello spazio dei giunti ..... | 49 |
| 5.2.2 Controllo a dinamica inversa nello spazio dei giunti .....              | 52 |
| 5.2.3 Controllo robusto .....   | 60 |
| 6. CONTROLLO NELLO SPAZIO OPERATIVO .....                                     | 69 |
| 6.1 Controllo PD con compensazione di velocità nello spazio operativo .....   | 69 |
| <i>Implementazione</i> .....  | 70 |
| 6.2 Controllo a dinamica inversa nello spazio operativo .....                 | 75 |
| <i>Implementazione</i> .....  | 75 |
| 7. CONCLUSIONI .....  | 80 |
| 8. RIFERIMENTI .....  | 82 |

# **Lista figure**

|  |    |
|--|----|
| Figura 1: Manipolatore planare a 2 link .....  | 6  |
| Figura 2: Distanze tra i giunti e i motori .....   | 7  |
| Figura 3: Variabili Matlab manipolatore .....  | 13 |
| Figura 4: Equazioni del moto Matlab .....  | 13 |
| Figura 5: Risoluzione equazioni in Matlab .....  | 14 |
| Figura 6: Modello dinamico del manipolatore in Simulink .....  | 14 |
| Figura 7: Elementi costanti del manipolatore in Matlab .....   | 15 |
| Figura 8: Implementazione traiettoria Simulink.....  | 16 |
| Figura 9: Step utilizzati per l'implementazione della traiettoria in Simulink .....                                | 17 |
| Figura 10: Plot traiettoria di accelerazione, velocità e posizione .....   | 18 |
| Figura 11: Caratterizzazione del sistema manipolatore e azionamenti .....  | 20 |
| Figura 12: Schema di controllo degli electric drives che guidano ogni giunto.....                                  | 21 |
| Figura 13: Struttura generale di controllo indipendente al giunto .....  | 22 |
| Figura 14: Implementazione dei disturbi d1 e d2 in Simulink .....  | 24 |
| Figura 15: Implementazione generale del controllo indipendente ai giunti in Simulink.....                          | 25 |
| Figura 16: Implementazione Simulink del feedback position control .....  | 26 |
| Figura 17: Simulazione feedback position control.....  | 27 |
| Figura 18: Implementazione Simulink del feedback position and velocity control .....                               | 28 |
| Figura 19: Simulazione feedback position and velocity control.....   | 29 |
| Figura 20: Implementazione Simulink del feedback position, velocity and acceleration control.....                  | 30 |
| Figura 21: Simulazione feedback position, velocity and acceleration control .....                                  | 31 |
| Figura 22: Implementazione con compensazione in avanti decentralizzata - posizione .....                           | 33 |
| Figura 23: Simulazione compensazione in avanti decentralizzata - posizione.....                                    | 34 |
| Figura 24: Implementazione con compensazione in avanti decentralizzata – posizione e velocità .....                | 35 |
| Figura 25: Simulazione compensazione in avanti decentralizzata - posizione e velocità .....                        | 36 |
| Figura 26: Implementazione con compensazione in avanti decentralizzata – posizione, velocità e accelerazione ..... | 37 |
| Figura 27: Simulazione compensazione in avanti decentralizzata - posizione, velocità e accelerazione .....         | 38 |
| Figura 28: Errore compensazione in avanti decentralizzata – posizione, velocità e accelerazione .....              | 39 |
| Figura 29: Schema di controllo a coppia precalcolata.....  | 40 |
| Figura 30: Blocco per implementazione coppia precalcolata .....  | 41 |
| Figura 31: Implementazione blocco dr1 .....  | 41 |
| Figura 32: Implementazione Simulink coppia precalcolata - posizione .....  | 42 |
| Figura 33: Simulazione coppia precalcolata - posizione .....   | 43 |
| Figura 34: Implementazione Simulink coppia precalcolata - posizione e velocità.....                                | 44 |
| Figura 35: Simulazione coppia precalcolata - posizione e velocità.....   | 45 |
| Figura 36: Implementazione Simulink coppia precalcolata - posizione, velocità e accelerazione .....                | 46 |
| Figura 37: Simulazione coppia precalcolata – posizione, velocità e accelerazione .....                             | 47 |
| Figura 38: Errore coppia precalcolata – posizione, velocità e accelerazione.....                                   | 47 |
| Figura 39: Schema di controllo PD ai giunti con compensazione di gravità .....                                     | 49 |
| Figura 40: Condizioni iniziali e postura costante desiderata .....   | 50 |
| Figura 41: Implementazione Simulink del controllo PD con compensazione di gravità nello spazio dei giunti .....    | 51 |
| Figura 42: Implementazione compensazione di gravità .....  | 51 |
| Figura 43: Simulazione controllo PD con compensazione di gravità nello spazio dei giunti .....                     | 52 |

|  |    |
|--|----|
| Figura 44: Schema di controllo a dinamica inversa ai giunti.....   | 53 |
| Figura 45: Implementazione Simulink del controllo a dinamica inversa nello spazio dei giunti .....             | 54 |
| Figura 46: Implementazione in Simulink della matrice inerziale $B(q)$ .....                                    | 55 |
| Figura 47: Implementazione in Simulink di $n(q, q_d)$ .....  | 55 |
| Figura 48: Implementazione in Simulink della matrice $F$ .....   | 56 |
| Figura 49: Implementazione in Simulink della matrice $C(q, q_d)$ .....   | 56 |
| Figura 50: Simulazione controllo a dinamica inversa nello spazio dei giunti - 1 .....                          | 57 |
| Figura 51: Simulazione controllo a dinamica inversa nello spazio dei giunti - 2.....                           | 58 |
| Figura 52: Errore della simulazione del controllo a dinamica inversa nello spazio dei giunti - 2 .....         | 59 |
| Figura 53: Schema di controllo robusto ai giunti .....   | 61 |
| Figura 54: Variabili controllo robusto Matlab.....   | 62 |
| Figura 55: Codice Matlab per determinazione matrice $Q$ – controllo robusto.....                               | 63 |
| Figura 56: Implementazione Simulink del controllo robusto .....  | 63 |
| Figura 57: Implementazione del blocco $B(q)$ .....   | 64 |
| Figura 58: Implementazione di $nq, q$ .....  | 64 |
| Figura 59: Implementazione della matrice dei coefficienti di smorzamento viscoso $F_v$ .....                   | 65 |
| Figura 60: Implementazione della robustezza in Simulink .....  | 65 |
| Figura 61: Funzione Matlab in Simulink che implementa $\omega$ .....   | 66 |
| Figura 62: Simulazione controllo robusto nello spazio dei giunti - 1 .....                                     | 66 |
| Figura 63: Simulazione controllo robusto nello spazio dei giunti - 2 .....                                     | 67 |
| Figura 64: Errore controllo robusto nello spazio dei giunti - 2.....   | 68 |
| Figura 65: Schema di controllo PD con compensazione di gravità nello spazio operativo.....                     | 69 |
| Figura 66: Rappresentazione grafica dello spazio raggiungibile dal manipolatore .....                          | 70 |
| Figura 67: Rappresentazione posizione iniziale e finale dell'end-effector .....                                | 71 |
| Figura 68: Implementazione Simulink del controllo PD con compensazione di gravità nello spazio operativo ..... | 71 |
| Figura 69: Implementazione Simulink della funzione di cinematica diretta $k()$ .....                           | 72 |
| Figura 70: Implementazione Simulink Jacobiano analitico .....  | 73 |
| Figura 71: Implementazione Simulink Jacobiano analitico trasposto .....  | 73 |
| Figura 72: Simulazione controllo PD con compensazione di gravità nello spazio operativo .....                  | 74 |
| Figura 73: Schema di controllo a dinamica inversa nello spazio operativo .....                                 | 75 |
| Figura 74: Implementazione Simulink a dinamica inversa nello spazio operativo .....                            | 76 |
| Figura 75: Implementazione Simulink inversa dello Jacobiano analitico .....                                    | 77 |
| Figura 76: Implementazione Simulink della derivata dello Jacobiano analitico .....                             | 77 |
| Figura 77: Simulazione controllo a dinamica inversa nello spazio operativo .....                               | 78 |

# 1. INTRODUZIONE

In questo documento sono presentate le varie tipologie di controllo del moto per un manipolatore planare a due gradi di libertà (2DOF).

Prima di tutto, si procede con lo studio del manipolatore al fine di individuare il suo modello dinamico, in presenza di due motori DC posizionati in concomitanza dei due giunti rotoidali.

Si prosegue con l'implementazione dei seguenti controlli di moto in Matlab e Simulink:

- 1) Controllo nello spazio dei giunti
  - a) Controllo Decentralizzato (Controllo indipendente ai giunti)
    - Feedback control
    - Feedforward compensation
    - Computed torque feedforward control
  - b) Controllo Centralizzato
    - Controllo PD con compensazione di gravità
    - Controllo a dinamica inversa
    - Controllo robusto
- 2) Controllo nello spazio operativo
  - Controllo PD con compensazione di gravità
  - Controllo a dinamica inversa

Infine, è stato fatto un confronto tra tutti i controlli implementati.

## 2. MODELLO DINAMICO DEL MANIPOLATORE

Si considera un manipolatore a due link con motori DC in figura [1]:

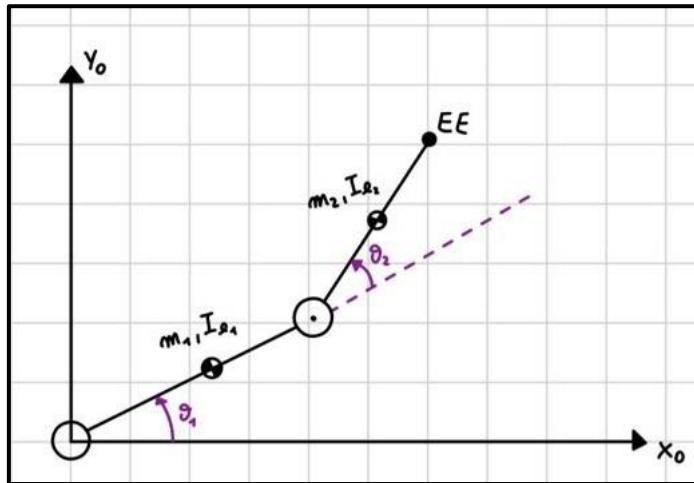


Figura 1: Manipolatore planare a 2 link

Il cui vettore delle variabili generalizzate è  $q = \begin{bmatrix} \vartheta_1 \\ \vartheta_2 \end{bmatrix}$ .

Si assumono le lunghezze dei link  $L_1$  e  $L_2$ , le cui masse sono  $m_1$  e  $m_2$ . I motori hanno rispettivamente massa  $m_{m1}$  e  $m_{m2}$ . Si considerano i motori posti sugli assi dei giunti con centri di massa coincidenti con l'origine dei frame dei giunti. Quindi, si definiscono  $d_1$  e  $d_2$  rispettivamente le distanze dei centri di massa dagli assi dei giunti.

Il modello dinamico è:

$$B(q) \ddot{q} + C(q, \dot{q}) \dot{q} + F_v \dot{q} + g(q) = \tau$$

Dove:

- $B(q)$ : è la matrice di inerzia;
- $C(q, \dot{q})$ : è una matrice che contiene i contributi di forza centrifuga e di Coriolis;
- $F_v$ : è la matrice diagonale dei coefficienti di attrito viscoso;
- $g(q)$ : sono i termini dipendenti dall'accelerazione gravitazionale;
- $\tau$ : sono le coppie di attuazione.

Si valuta:

$$B(q) = \sum_{i=1}^n \{m_i [J_P]^T J_P + [J_O]^T R_i I_i R_i^T J_O + m_{mi} [J_P^m]^T J_P^m + [J_O^m]^T R_{mi} I_{mi} R_{mi}^T J_O^m\} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

Dove:

- $J_P$  e  $J_P^m$ : Jacobiani di posizione rispettivamente del link e del motore
- $J_O$  e  $J_O^m$ : Jacobiani di orientamento rispettivamente del link e del motore
- $I_i$  e  $I_{mi}$ : matrici inerziali rispettivamente del link e del motore

- $R_i$  e  $R_{mi}$ : matrici di rotazione.

Possiamo valutare gli Jacobiani dei link e dei motori in modo grafico. Infatti, dalla figura [2] si possono notare le relative distanze, e quindi posizioni:

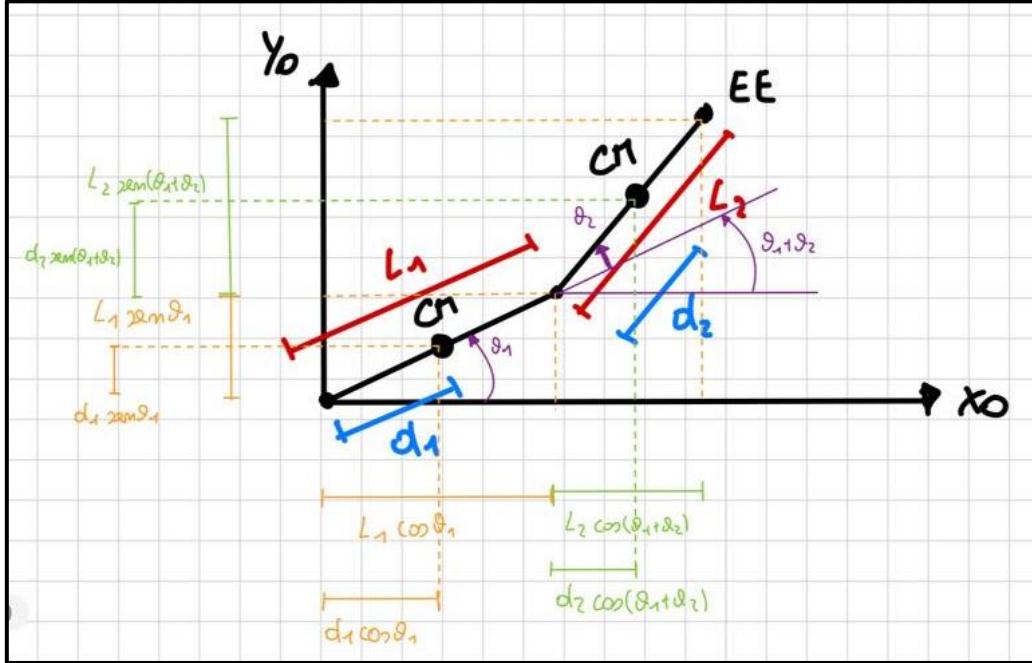


Figura 2: Distanze tra i giunti e i motori

Gli Jacobiani dei due link sono:

$$J_1 = \begin{bmatrix} -d_1 s_1 & 0 \\ d_1 c_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad J_{P1} \quad J_{O1}$$

$$J_2 = \begin{bmatrix} -L_1 s_1 - d_2 s_{12} & -d_2 s_{12} \\ L_1 c_1 + d_2 c_{12} & d_2 c_{12} \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \quad J_{P2} \quad J_{O2}$$

Dove  $c_1, s_1, c_{12}, s_{12}$  indicano rispettivamente  $\cos(\vartheta_1)$ ,  $\sin(\vartheta_1)$ ,  $\cos(\vartheta_1 + \vartheta_2)$ ,  $\sin(\vartheta_1 + \vartheta_2)$ .

Per quanto riguarda i motori, si suppone di avere una trasmissione rigida con rapporto di trasmissione  $k_{ri}$ .

Il primo rotore si trova in posizione  $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ , la rotazione avviene solo rispetto a  $z_0$  con rapporto di trasmissione  $k_{r1}$ :

$$J^{m1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ k_{r1}z_{m1} & 0 \end{bmatrix}$$

*J<sub>P</sub><sup>m1</sup>*

*J<sub>O</sub><sup>m1</sup>*

Con  $z_{m1}$  il versore dell'asse di rotazione del rotore, le cui componenti sono riferite alla terna base (=1).

Per il secondo rotore valutiamo la distanza dalla terna base (dal frame 1 al frame 0). La rotazione avviene lungo l'asse  $z_1$  con rapporto di trasmissione  $k_{r2}$ :

$$J^{m2} = \begin{bmatrix} -L_1 s_1 & 0 \\ L_1 c_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & k_{r2} \end{bmatrix}$$

*J<sub>P</sub><sup>m2</sup>*

*J<sub>O</sub><sup>m2</sup>*

Gli ultimi elementi utili per valutare la matrice d'inerzia  $B(q)$ , sono le inerzie dei due link e dei due motori. Per facilitare il calcolo di  $B(q)$ , consideriamo l'inerzia generale:

$$I_{li} = \begin{bmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{bmatrix} \quad I_{mi} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}$$

Dove con  $I_{li}$  si indica l'inerzia del link i-esimo e con  $I_{mi}$  del motore i-esimo. Quest'ultime sono diagonali in quanto il rotore gira attorno al proprio asse.

Quindi, riconsiderando l'espressione matematica di  $B(q)$ , avremo due equazioni:

$$\text{per } i = 1: \quad m_1 \begin{bmatrix} -d_1 s_1 & d_1 c_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -d_1 s_1 & 0 \\ d_1 c_1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \\ + m_{m1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & k_{r1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_{r1} & 0 \end{bmatrix}$$

$$\text{per } i = 2: \quad m_2 \begin{bmatrix} -L_1 s_1 - d_2 s_{12} & L_1 c_1 + d_2 c_{12} & 0 \\ -d_2 s_{12} & d_2 c_{12} & 0 \end{bmatrix} \begin{bmatrix} -L_1 s_1 - d_2 s_{12} & -d_2 s_{12} \\ L_1 c_1 + d_2 c_{12} & d_2 c_{12} \\ 0 & 0 \end{bmatrix} + \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} + m_{m2} \begin{bmatrix} -L_1 s_1 & L_1 c_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -L_1 s_1 & 0 \\ L_1 c_1 & 0 \\ 0 & 0 \end{bmatrix} + \\ + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & k_{r2} \end{bmatrix} \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & k_{r2} \end{bmatrix}$$

Effettuando i calcoli e sommando si ottiene:

$$B(q) = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \\ \begin{bmatrix} I_1 + m_1 d_1^2 + k_{r1}^2 I_{m1} + I_2 + m_2 (L_1^2 + d_2^2 + 2L_1 d_2 c_2) + I_{m2} + m_{m2} L_1^2 & I_2 + m_2 (d_2^2 + L_1 d_2 c_2) + k_{r2} I_{m2} \\ I_2 + m_2 (d_2^2 + L_1 d_2 c_2) + k_{r2} I_{m2} & I_2 + m_2 d_2^2 + k_{r2} I_{m2} \end{bmatrix}$$

Dove:

$$I_1 = I_{33}^{(1)}, \quad I_2 = I_{33}^{(2)}, \quad I_{m1} = I_3^{(m1)}, \quad I_{m2} = I_3^{(m2)}$$

Per valutare la matrice  $C(q, \dot{q}) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$  si devono prima valutare i coefficienti  $c_{ijk}$ , detti coefficienti di Christoffel.

Il generico  $c_{ij}$  risulta:

$$c_{ij} = \sum_{k=1}^n c_{ijk} \dot{q}_k$$

Dove, i coefficienti  $c_{ijk}$  sono:

$$c_{ijk} = \frac{1}{2} \left( \frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right) \quad \text{con } c_{ijk} = c_{ikj}$$

Quindi, i coefficienti di Christoffel sono:

$$\begin{aligned} c_{111} &= \frac{1}{2} \frac{\partial b_{11}}{\partial q_1} = 0; & c_{112} &= \frac{1}{2} \frac{\partial b_{11}}{\partial q_2} = -m_2 L_1 d_2 s_2 = c_{121} \\ c_{122} &= \frac{\partial b_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial b_{22}}{\partial q_1} = -m_2 L_1 d_2 s_2; & c_{211} &= \frac{\partial b_{21}}{\partial q_1} - \frac{1}{2} \frac{\partial b_{11}}{\partial q_2} = m_2 L_1 d_2 s_2 \\ c_{212} &= \frac{1}{2} \frac{\partial b_{22}}{\partial q_1} = 0 = c_{221}; & c_{222} &= \frac{1}{2} \frac{\partial b_{22}}{\partial q_2} = 0 \end{aligned}$$

Quindi, gli elementi della matrice  $C(q, \dot{q})$  sono:

$$c_{11} = c_{111} * \dot{q}_1 + c_{112} * \dot{q}_2 = -m_2 L_1 d_2 s_2 * \dot{\vartheta}_2$$

$$c_{22} = c_{221} * \dot{q}_1 + c_{222} * \dot{q}_2 = 0$$

$$c_{12} = c_{121} * \dot{q}_1 + c_{122} * \dot{q}_2 = -m_2 L_1 d_2 s_2 * \dot{\vartheta}_1 - m_2 L_1 d_2 s_2 * \dot{\vartheta}_2 = -m_2 L_1 d_2 s_2 * (\dot{\vartheta}_1 + \dot{\vartheta}_2)$$

$$c_{21} = c_{211} * \dot{q}_1 + c_{212} * \dot{q}_2 = m_2 L_1 d_2 s_2 * \dot{\vartheta}_1$$

Tale che,  $C(q, \dot{q})$  risulta:

$$C(q, \dot{q}) = \begin{bmatrix} -m_2 L_1 d_2 s_2 * \dot{\vartheta}_2 & -m_2 L_1 d_2 s_2 * (\dot{\vartheta}_1 + \dot{\vartheta}_2) \\ m_2 L_1 d_2 s_2 * \dot{\vartheta}_1 & 0 \end{bmatrix}$$

È ora necessario procedere con il calcolo del vettore di gravità  $g(q)$ :

$$g_i(q) = \sum_{j=1}^n [m_j g_0^T J_{P_i}(q) + m_{mj} g_0^T J_{P_i}^{(m)}(q)] \quad \text{con } g_0 = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix}$$

Quindi:

per  $i = 1$ :

$$\begin{aligned} g_1(q) = & -m_1 [0 \quad -g \quad 0] \begin{bmatrix} -d_1 s_1 \\ d_1 c_1 \\ 0 \end{bmatrix} - m_{m1} [0 \quad -g \quad 0] \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - m_2 [0 \quad -g \quad 0] \begin{bmatrix} -L_1 s_1 - d_2 s_{12} \\ L_1 c_1 + d_2 c_{12} \\ 0 \end{bmatrix} \\ & - m_{m2} [0 \quad -g \quad 0] \begin{bmatrix} -L_1 s_1 \\ L_1 c_1 \\ 0 \end{bmatrix} \end{aligned}$$

per  $i = 2$ :

$$\begin{aligned} g_2(q) = & -m_1 [0 \quad -g \quad 0] \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - m_{m1} [0 \quad -g \quad 0] \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - m_2 [0 \quad -g \quad 0] \begin{bmatrix} -d_2 s_{12} \\ d_2 c_{12} \\ 0 \end{bmatrix} \\ & - m_{m2} [0 \quad -g \quad 0] \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

Effettuando i calcoli si ottiene:

$$g(q) = \begin{bmatrix} (m_1 d_1 + m_2 L_1 + m_{m2} L_1) * g c_1 + m_2 d_2 g c_{12} \\ m_2 d_2 g c_{12} \end{bmatrix}$$

Infine, si suppone la presenza di attrito viscoso:  $F_v = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix}$ , dove con  $b_i$  indichiamo il coefficiente di smorzamento viscoso.

## 2.1 Equazioni del moto del manipolatore

Trovate tutte le matrici, è possibile utilizzare l'espressione del modello dinamico del manipolatore per trovare le due equazioni di moto.

Si ha:

$$\begin{bmatrix} I_1 + m_1 d_1^2 + k_{r1}^2 I_{m1} + I_2 + m_2(L_1^2 + d_2^2 + 2L_1 d_2 c_2) + I_{m2} + m_{m2} L_1^2 & I_2 + m_2(d_2^2 + L_1 d_2 c_2) + k_{r2} I_{m2} \\ I_2 + m_2(d_2^2 + L_1 d_2 c_2) + k_{r2} I_{m2} & I_2 + m_2 d_2^2 + k_{r2} I_{m2} \end{bmatrix} * \begin{bmatrix} \ddot{\vartheta}_1 \\ \ddot{\vartheta}_2 \end{bmatrix} + \begin{bmatrix} -m_2 L_1 d_2 s_2 * \dot{\vartheta}_2 & -m_2 L_1 d_2 s_2 * (\dot{\vartheta}_1 + \dot{\vartheta}_2) \\ m_2 L_1 d_2 s_2 * \dot{\vartheta}_1 & 0 \end{bmatrix} * \begin{bmatrix} \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \end{bmatrix} + \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix} * \begin{bmatrix} \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \end{bmatrix} + \begin{bmatrix} (m_1 d_1 + m_2 L_1 + m_{m2} L_1) * g c_1 + m_2 d_2 g c_{12} \\ m_2 d_2 g c_{12} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

La prima equazione di moto è:

$$(I_1 + m_1 d_1^2 + k_{r1}^2 I_{m1} + I_2 + m_2(L_1^2 + d_2^2 + 2L_1 d_2 c_2) + I_{m2} + m_{m2} L_1^2) * \ddot{\vartheta}_1 + (I_2 + m_2(d_2^2 + L_1 d_2 c_2) + k_{r2} I_{m2}) * \ddot{\vartheta}_2 - 2m_2 L_1 d_2 s_2 * \dot{\vartheta}_1 * \dot{\vartheta}_2 - m_2 L_1 d_2 s_2 * \dot{\vartheta}_2^2 + (m_1 d_1 + m_2 L_1 + m_{m2} L_1) * g c_1 + m_2 d_2 g c_{12} + b_1 * \dot{\vartheta}_1 = \tau_1$$

La seconda equazione di moto è:

$$(I_2 + m_2(d_2^2 + L_1 d_2 c_2) + k_{r2} I_{m2}) * \ddot{\vartheta}_1 + (I_2 + m_2 d_2^2 + k_{r2}^2 I_{m2}) * \ddot{\vartheta}_2 + m_2 L_1 d_2 s_2 * \dot{\vartheta}_1^2 + m_2 d_2 g c_{12} + b_2 * \dot{\vartheta}_2 = \tau_2$$

È inoltre possibile andare ad esplicitare le coppie ai giunti  $\tau_i$ :

$$\tau = K_r K_t R_a^{-1} G_v v_c - K_r K_t R_a^{-1} K_v K_r \dot{q}$$

Quindi, imponendo  $F = F_v + K_r K_t R_a^{-1} K_v K_r$  e  $u = K_r K_t R_a^{-1} G_v v_c$ , dove  $u$  è l'input del sistema manipolatore, si ottiene il seguente modello dinamico con le conseguenti nuove equazioni di moto:

$$B(q) \ddot{q} + C(q, \dot{q}) \dot{q} + F_v \dot{q} + g(q) = \tau$$

Le equazioni sono:

$$(I_1 + m_1 d_1^2 + k_{r1}^2 I_{m1} + I_2 + m_2(L_1^2 + d_2^2 + 2L_1 d_2 c_2) + I_{m2} + m_{m2} L_1^2) * \ddot{\vartheta}_1 + (I_2 + m_2(d_2^2 + L_1 d_2 c_2) + k_{r2} I_{m2}) * \ddot{\vartheta}_2 - 2m_2 L_1 d_2 s_2 * \dot{\vartheta}_1 * \dot{\vartheta}_2 - m_2 L_1 d_2 s_2 * \dot{\vartheta}_2^2 + (m_1 d_1 + m_2 L_1 + m_{m2} L_1) * g c_1 + m_2 d_2 g c_{12} + \left( b_1 + \frac{k_{t1} k_{v1} k_{r1}^2}{R_{a1}} \right) \dot{\vartheta}_1 = u_1$$

$$(I_2 + m_2(d_2^2 + L_1 d_2 c_2) + k_{r2} I_{m2}) * \ddot{\vartheta}_1 + (I_2 + m_2 d_2^2 + k_{r2}^2 I_{m2}) * \ddot{\vartheta}_2 + m_2 L_1 d_2 s_2 * \dot{\vartheta}_1^2 \\ + m_2 d_2 g c_{12} + \left( b_2 + \frac{k_{t2} k_{v2} k_{r2}^2}{R_{a2}} \right) \dot{\vartheta}_2 = u_2$$

Dove:

$R_{ai}$ : resistenza di armatura del motore i-esimo;

$k_{ti}$ : costante di coppia del motore i-esimo;

$k_{vi}$ : costante di tensione del motore i-esimo;

## 2.2 Implementazione Matlab e Simulink

Per poter rappresentare il manipolatore come modello in Simulink si sono prima di tutto definite le variabili in Matlab:

```
%> Variabili
syms m1 m2 %Masse link
syms L1 L2 %Lunghezze link
syms I1 I2 %Inerzie link

syms d1 d2 %Distanze Joint-CM link

syms g %Gravità

syms th1 th2 %Posizione
syms th1d th2d %Velocità
syms th1dd th2dd %Accelerazione

syms mm1 mm2 %Masse rotori
syms Im1 Im2 %Inerzie rotori
syms Ra1 Ra2 %Resistenze armatura
syms kt1 kt2 %Costanti di coppia
syms kv1 kv2 %Costanti di tensione

syms kr1 kr2 %Rapporti di trasmissione

syms b1 b2 %Coeffienti di attrito viscoso

syms u1 u2 %Control input
```

Figura 3: Variabili Matlab manipolatore

Di seguito sono state scritte le equazioni del moto per poterle successivamente risolvere con la funzione [solve](#):

|  |  |
|--|--|
| <pre>%% Equazioni di moto Eq1 = (I1 + m1*d1^2 + Im1*kr1^2 + I2 + Im2 + mm2*L1^2 + m2*(L1^2 + d2^2 + 2*L1*d2*cos(th2)))*th1dd + ... + (I2 + kr2*Im2 + m2*(d2^2 + L1*d2*cos(th2)))*th2dd - 2*m2*L1*d2*sin(th2)*th1d*th2d - m2*L1*d2*sin(th2)*th2d*th2d + ... + g*(m1*d1 + mm2*L1 + m2*L1)*cos(th1) + m2*d2*g*cos(th1+th2) + (b1 + kr1*kt1*kv1*kr1*(1/Ra1))*th1d == u1;  Eq2 = (I2 + kr2*Im2 + m2*(d2^2 + L1*d2*cos(th2)))*th1dd + (I2 + m2*d2^2 + Im2*kr2^2)*th2dd + ... + m2*L1*d2*sin(th2)*th1d*th1d + m2*d2*g*cos(th1+th2) + (b2 + kr2*kt2*kv2*kr2*(1/Ra2))*th2d == u2;</pre> |  |
| <pre>%% Risoluzione equazioni S = solve([Eq1, Eq2], [th1dd, th2dd]);</pre>   |  |
| <pre>TH1DD = S.th1dd; TH2DD = S.th2dd;</pre>   |  |

Figura 4: Equazioni del moto Matlab

Infine, per creare il modello Simulink del manipolatore utilizzando le equazioni del moto risolte rispetto a  $\ddot{\vartheta}_1$  e  $\ddot{\vartheta}_2$ , è stata utilizzata la seguente parte di codice:

```

%% Risolviamo le equazioni con Simulink
TF_Sim = true;
if(true == TF_Sim)
    MODEL1 = 'EOM_SIM1';
    %Usiamo exist=4 per verificare che esista un modello
    %Simulink nel folder.
    if(4 == exist(MODEL1))
        close_system(MODEL1, 0); %Chiudo il sistema senza salvare
        delete(MODEL1);
    end
    new_system(MODEL1)
    open_system(MODEL1)

    %Mettiamo entrambe le equazioni nella funzione Matlab in Simulink
    matlabFunctionBlock( [MODEL1,'/Thetadd_eq'], TH1DD, TH2DD, ...
        'Optimize', false, ...           %Per non far semplificare le espressioni da Simulink
        'Outputs', {'theta1_DD', 'theta2_DD'});
end

```

Figura 5: Risoluzione equazioni in Matlab

Di seguito il modello Simulink del manipolatore:

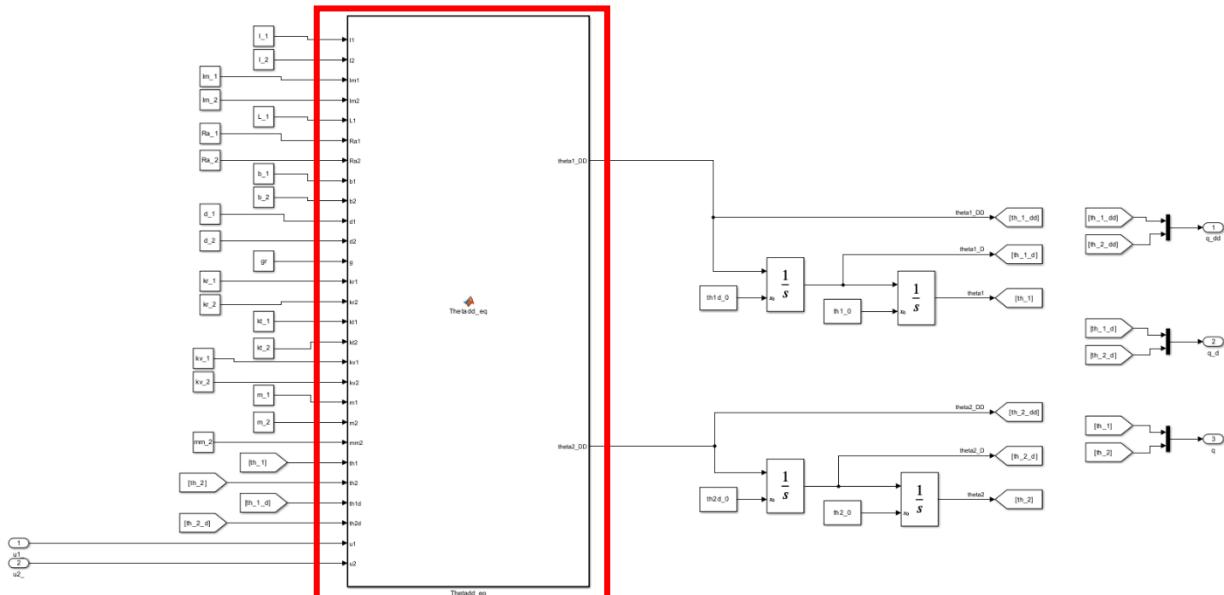


Figura 6: Modello dinamico del manipolatore in Simulink

Il blocco evidenziato di rosso è quello che contiene le equazioni di moto del manipolatore. Le uniche due uscite sono le accelerazioni angolari  $\ddot{\vartheta}_1$  e  $\ddot{\vartheta}_2$ . Utilizzando due integratori (con condizioni iniziali  $\dot{\vartheta}_1 = 0 \text{ rad/s}$ ,  $\vartheta_1 = 30^\circ$ ,  $\dot{\vartheta}_2 = 0 \text{ rad/s}$ ,  $\vartheta_2 = 45^\circ$ ) rispettivamente per i due giunti, si ottengono la velocità angolare e la posizione angolare dei giunti nel tempo.

Gli ingressi presentano i parametri fissi del manipolatore e gli input di controllo  $u_1$  e  $u_2$  che dipendono dal tipo di controllo utilizzato.

I parametri fissi sono:

- $L_1 = 1$ ,  $L_2 = 1$ : Lunghezze link (m)
- $d_1 = 0.5$ ,  $d_2 = 0.5$ : Distanze dei baricentri dai giunti (m)
- $m_1 = 50$ ,  $m_2 = 50$ : Masse link (Kg)
- $I_1 = 10$ ,  $I_2 = 10$ : Inerzie link ( $\text{Kg} \cdot \text{m}^2$ )
- $m_{m1} = 5$ ,  $m_{m2} = 5$ : Masse rotori (Kg)

- $I_{m1} = 0.01$ ,  $I_{m2} = 0.01$ : Inerzie rotori ( $\text{Kg} \cdot \text{m}^2$ )
- $k_{r1} = 100$ ,  $k_{r2} = 100$ : Rapporti di trasmissione
- $gr = 9.81$ : Gravità ( $\text{m}/\text{s}^2$ )
- $b_1 = 0.01$ ,  $b_2 = 0.01$ : Coefficienti di attrito viscoso del motore
- $R_{a1} = 10$ ,  $R_{a2} = 10$ : Resistenze di armatura ( $\Omega$ )
- $k_{t1} = 2$ ,  $k_{t2} = 2$ : Costanti di coppia ( $2 \text{ N} \cdot \text{m}/\text{A}$ )
- $k_{v1} = 2$ ,  $k_{v2} = 2$ : Costanti di tensione ( $2 \text{ V} \cdot \text{s}/\text{rad}$ )

I due bracci sono stati scelti uguali per evidenziare i contributi di interazione dinamica tra i due giunti.

## 2.3 Valori costanti per simulink

Per poter semplificare l'implementazione degli algoritmi di controllo su Simulink, sono stati definiti su Matlab gli elementi costanti del modello dinamico del manipolatore.

Come si vedrà in seguito, questi sono stati utilizzati per implementare i disturbi nel controllo indipendente ai giunti, il blocco di compensazione di gravità, i blocchi per la dinamica inversa e gli Jacobiani.

```
%> Elementi costanti

%g(q)
g1_1 = gr*(m_1*d_1 + mm_2*L_1 + m_2*L_1); % *cos(th1)
g1_2 = m_2*d_2*gr; % *cos(th(1+2))
g2 = m_2*d_2*gr; % *cos(th(1+2))

%B(q)
b11 = I_1 + m_1*d_1^2 + Im_1*kr_1^2 + I_2 + Im_2 + mm_2*L_1^2 + m_2*(L_1^2 + d_2^2);
b22 = I_2 + m_2*d_2^2 + Im_2*kr_2^2;
b12 = I_2 + kr_2*Im_2 + m_2*d_2^2;

delta_b11 = 2*m_2*L_1*d_2; % *cos(th2)
delta_b12 = m_2*L_1*d_2; % *cos(th2)

%c(q,qd)
h = -m_2*L_1*d_2; % *sin(th2)*th...

%F
F1 = b_1 + kr_1*kt_1*kv_1*kr_1*(1/Ra_1);
F2 = b_2 + kr_2*kt_2*kv_2*kr_2*(1/Ra_2);
```

Figura 7: Elementi costanti del manipolatore in Matlab

### 3. RIFERIMENTO E TRAIETTORIA

L'obiettivo di alcuni algoritmi di controllo è quello di far inseguire una traiettoria al manipolatore. Si impone quindi l'obiettivo di portare la variabile angolare  $q$  del giunto i-esimo da un valore iniziale  $q_i$  a un valore finale  $q_f$  in un intervallo di tempo  $t_f$ .

Un tipo di traiettoria che consente di raggiungere la configurazione finale voluta nel minor tempo possibile, senza avere discontinuità, è la traiettoria a profilo trapezoidale di velocità.

Un profilo trapezoidale di velocità impone un'accelerazione costante nella fase di partenza, una velocità di crociera, e una decelerazione costante nella fase di arrivo. La traiettoria che ne risulta è costituita da un tratto lineare raccordato con due tratti parabolici nell'intorno delle posizioni iniziale e finale.

Questa traiettoria è stata utilizzata come riferimento da far seguire al manipolatore nel controllo indipendente ai giunti, nel controllo a dinamica inversa e nel controllo robusto nello spazio dei giunti

I parametri utilizzati sono i seguenti:

- $t_c = 0.6$ : tempo in cui si conclude il tratto parabolico (s)
- $\dot{q}_c = 1$ : velocità di crociera/velocità max (rad/s)
- $q_i = 0, q_f = 1$ : posizione iniziale e finale (rad)
- $t_f = t_c + \frac{(q_f - q_i)}{\dot{q}_c}$ : tempo finale (s)
- $\ddot{q}_c = \frac{\dot{q}_c}{t_c} =$ : accelerazione massima (rad/s<sup>2</sup>)

La traiettoria trapezoidale è stata implementata su Simulink attraverso l'utilizzo dei blocchi *Step*, ottenendo dapprima la rappresentazione dell'accelerazione. Collegando a cascata due integratori si ottiene la rappresentazione della posizione angolare di riferimento (con il primo integratore, invece, si ottiene quella della velocità).

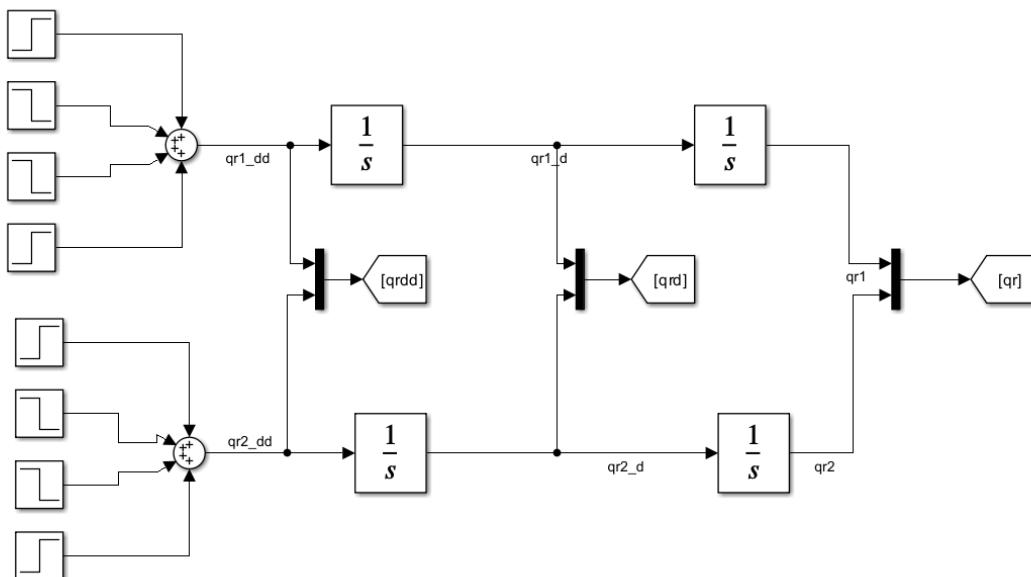


Figura 8: Implementazione traiettoria Simulink

Con i 4 step si è costruita l'accelerazione:

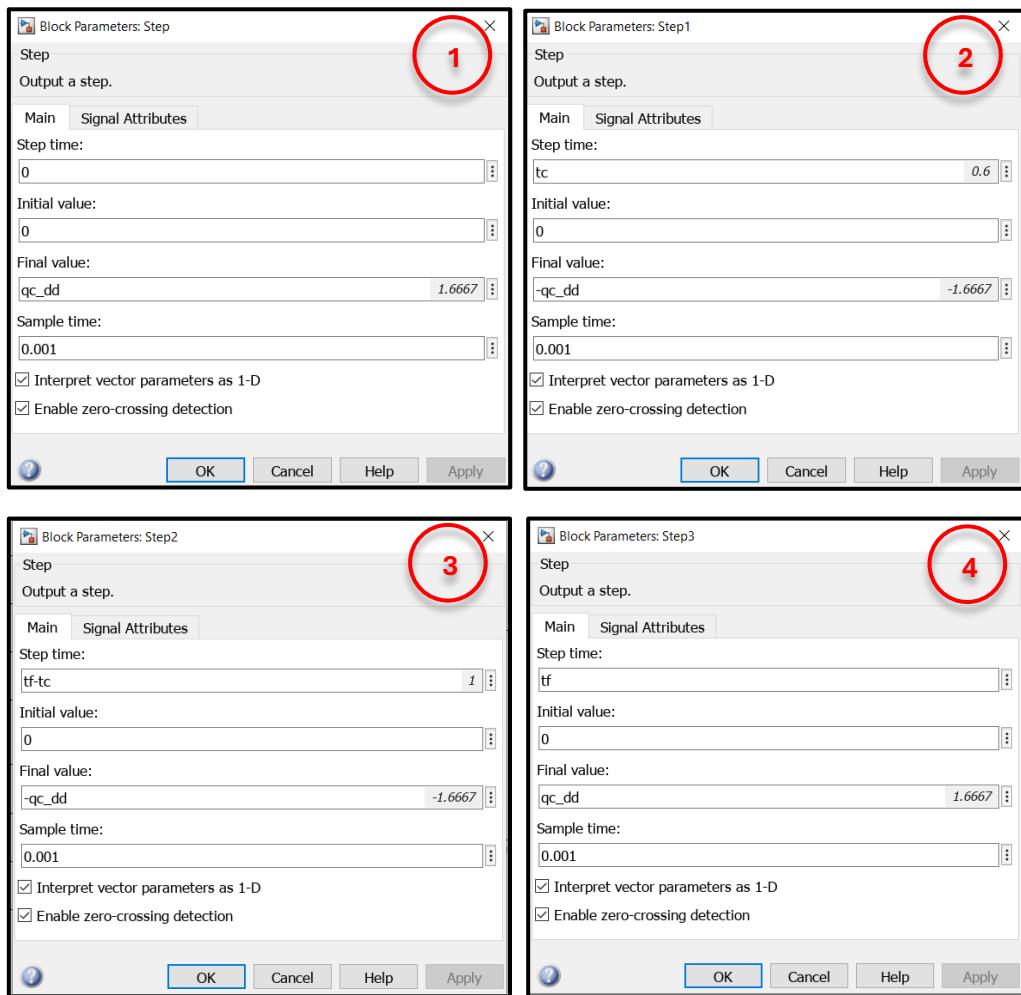


Figura 9: Step utilizzati per l'implementazione della traiettoria in Simulink

È stato utilizzato un tempo di campionamento di 1ms.

Di seguito i plot, rispettivamente, di accelerazione, velocità e posizione:

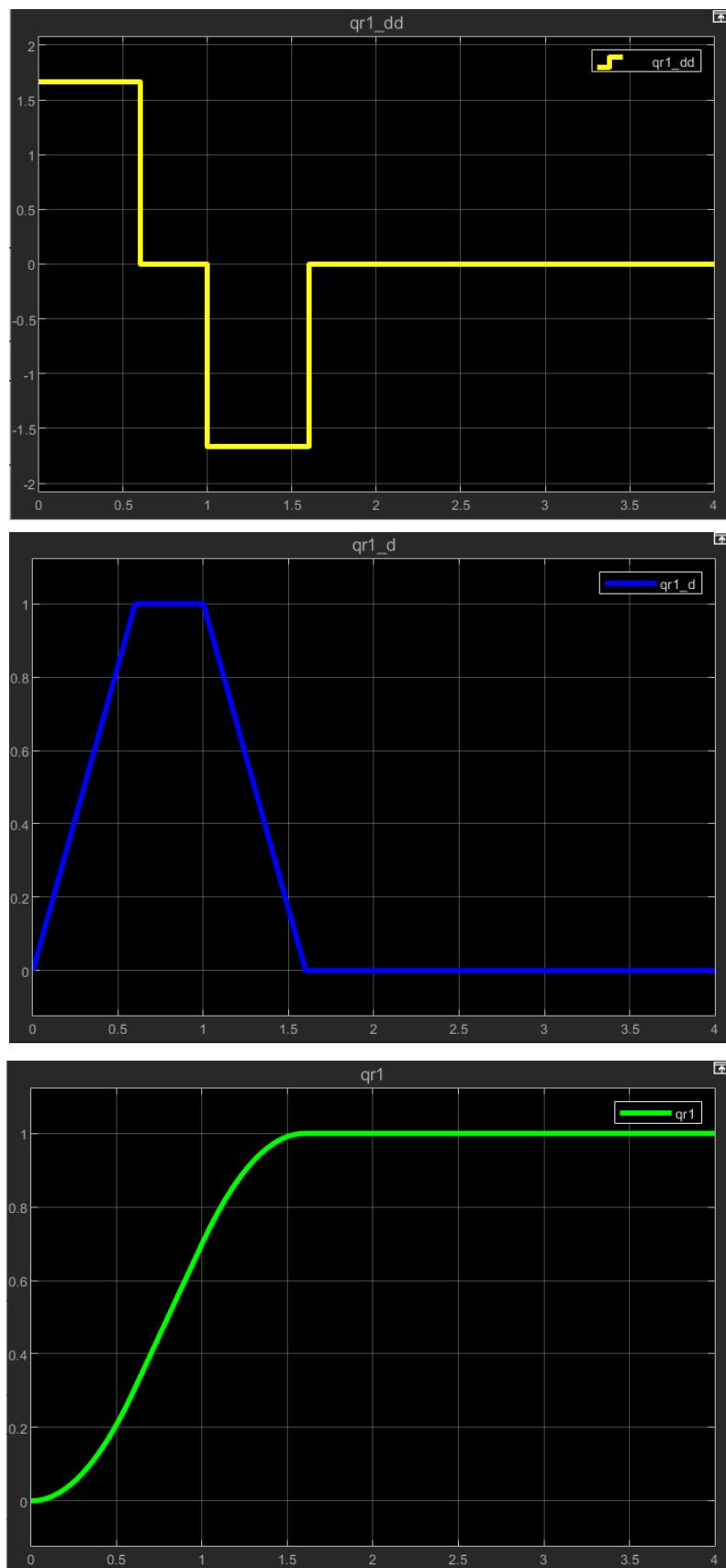


Figura 10: Plot traiettoria di accelerazione, velocità e posizione

Per gli algoritmi di controllo che non richiedono una traiettoria di riferimento da far inseguire al manipolatore, è stato definito un vettore di variabili  $q_r$ . Infatti, per il controllo PD con compensazione di gravità è richiesto che il manipolatore rimanga in una posa costante.

## 4. SINGOLARITÀ

Valutiamo le singolarità del manipolatore. Consideriamo solo le componenti  $\dot{p}_x$  e  $\dot{p}_y$  dello Jacobiano geometrico del manipolatore:

$$J = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix}$$

Per valutarne le singolarità, poniamo uguale a zero il determinante dello Jacobiano (imponiamo  $L_1 = L_2 = 1 \neq 0$ ) :

$$|J| = 0 \Rightarrow -s_1 c_{12} + c_1 s_{12} = 0 \Rightarrow \text{sviluppando } s_{12} \text{ e } c_{12} \Rightarrow s_2 = 0$$

Quindi, il manipolatore si trova in una postura singolare quando  $\vartheta_2 = 0, \pi$ , mentre  $\vartheta_1$  non è influente. Queste configurazioni si riferiscono ai casi in cui l'organo terminale del manipolatore è situato al confine esterno ( $\vartheta_2 = 0$ ) o interno ( $\vartheta_2 = \pi$ ) dello spazio di lavoro raggiungibile.

Si può concludere dicendo che non avrebbe senso far tendere il manipolatore verso una posizione che naturalmente non potrebbe ottenere.

Un esempio è effettuato nel paragrafo dove è presentato il controllo a dinamica inversa nello spazio operativo.

# 5. CONTROLLO NELLO SPAZIO DEI GIUNTI

## 5.1 Controllo Decentralizzato

Il controllo decentralizzato è l'opzione di controllo più basilare per un manipolatore, esso consiste nel trattare ogni giunto come un sistema SISO indipendente. Quindi si considera il manipolatore costituito da  $n$  sistemi indipendenti (gli  $n$  azionamenti di giunto) e gli effetti di accoppiamento tra i giunti, dovuti alla configurazione e al moto del manipolatore, sono visti come disturbi per gli azionamenti dei singoli giunti.

Questa semplificazione è fattibile nel caso in cui siamo in presenza di elevati rapporti di trasmissione  $K_r$  e se le traiettorie desiderate non richiedono prestazioni esigenti in termini di velocità e accelerazione.

Considerando il sistema a blocchi seguente:

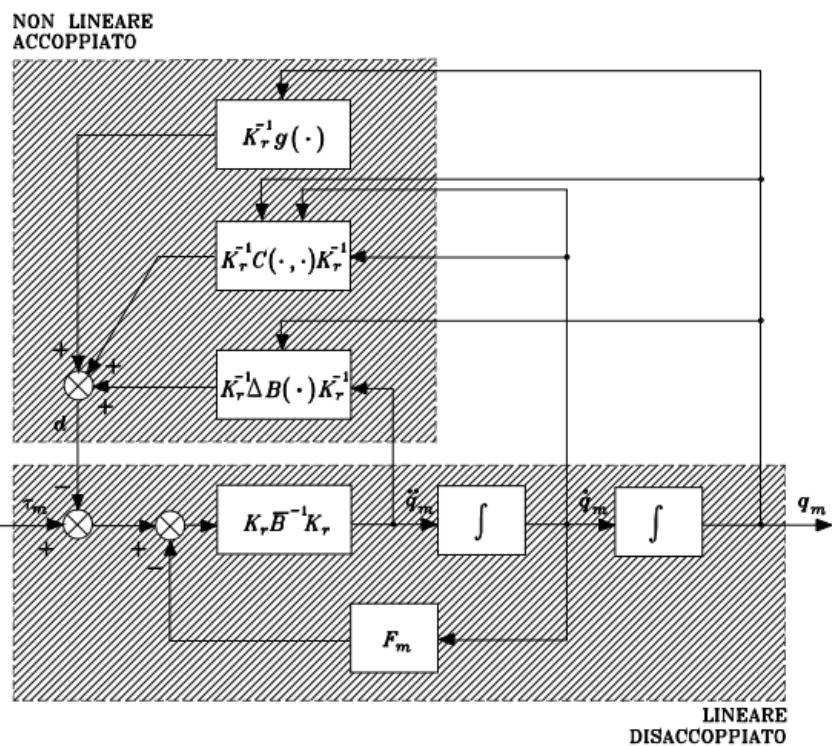


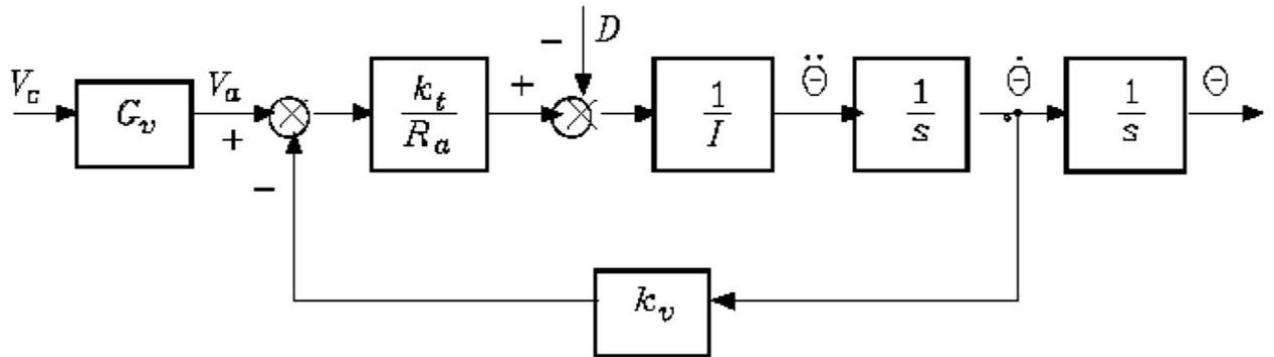
Figura 11: Caratterizzazione del sistema manipolatore e azionamenti

$$d = K_r^{-1} \Delta B(q) K_r^{-1} \ddot{q}_m + K_r^{-1} C(q, \dot{q}) K_r^{-1} \dot{q}_m + K_r^{-1} g(q)$$

$$\tau_m = K_r^{-1} \bar{B} K_r^{-1} \ddot{q}_m + F_m \dot{q}_m + d$$

Si nota che ciascun elemento di  $\tau_m$  influenza esclusivamente la componente corrispondente di  $q_m$ , mentre la variabile  $d$  tiene conto dei termini di accoppiamento dei giunti.

Lo schema di controllo degli electric drives che guidano ogni giunto è il seguente:



*Figura 12: Schema di controllo degli electric drives che guidano ogni giunto*

Si devono considerare le seguenti ipotesi:

$$\frac{L_a}{R_a} \ll \frac{I}{F_m} \quad ; \quad F_m \ll \frac{k_v k_t}{R_a}$$

Ovvero, la costante di tempo elettrica deve risultare trascurabile rispetto a quella meccanica e il coefficiente di attrito viscoso deve risultare trascurabile rispetto a quello elettrico.

Inoltre, si vuole assicurare la reiezione del disturbo  $d$  con un elevato valore del guadagno degli amplificatori a monte del punto di applicazione del disturbo e la presenza, nel controllore, di un'azione integrale. Questo suggerisce l'utilizzo di un controllore proporzionale-integrale(PI).

Si possono definire tre tipi schemi di controllo:

- Feedback control
- Feedforward compensation
- Computed torque feedforward control

E ognuno di esso può essere caratterizzato da un:

- Controllo di posizione
- Controllo di posizione e di velocità
- Controllo di posizione, velocità e accelerazione

### 5.1.1 Implementazione controllo decentralizzato

La struttura generale di controllo, che è caratterizzato dal controllo di posizione  $C_P(s)$ , velocità  $C_V(s)$  e accelerazione  $C_A(s)$  è la seguente:

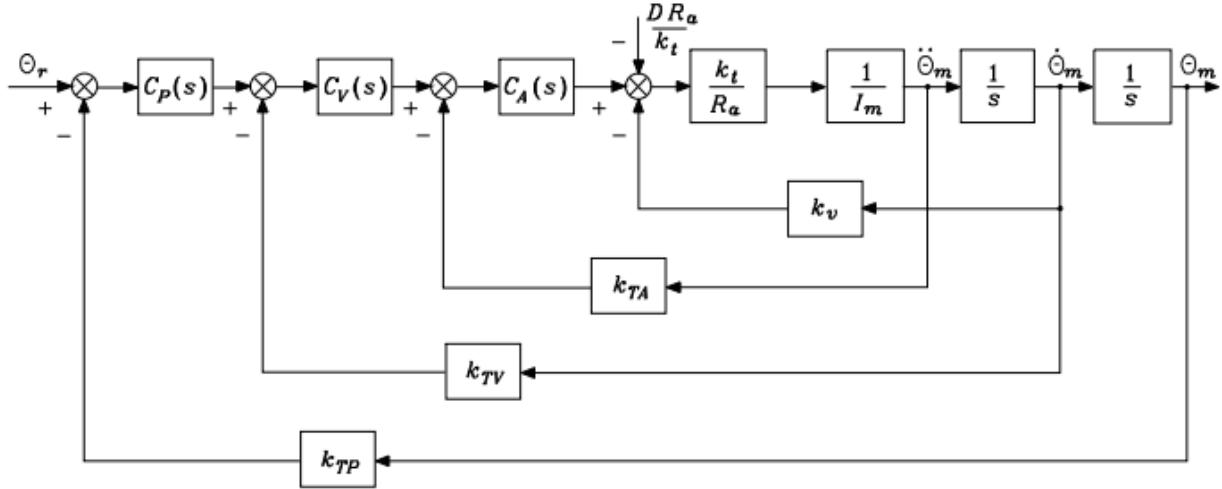


Figura 13: Struttura generale di controllo indipendente al giunto

Dove  $k_{TP}$  ,  $k_{TV}$  ,  $k_{TA}$  sono rispettivamente le costanti di trasduzione di posizione, velocità e accelerazione.

Per gli schemi di controllo in cui è presente il solo controllo di posizione l'azione di controllo sarà caratterizzata dal solo controllore *PI* di posizione:

$$C_P(s) = K_P \frac{(1 + sT_P)}{s} \quad C_V = C_A = 1 \quad k_{TV} = k_{TA} = 0$$

sono stati scelti i seguenti parametri:

$$K_P = 5 \quad T_P = 50 * T_m \quad k_{TP} = 1$$

Si è scelto  $T_P \gg T_m$  per assicurare l'asintotica stabilità del sistema, ove  $T_m$  è la costante di tempo del motore.

Per gli schemi di controllo caratterizzati dal controllo di posizione e velocità l'azione controllo sarà caratterizzato dal controllore *P* di posizione e dal controllore *PI* di velocità:

$$C_V(s) = K_V \frac{(1 + sT_V)}{s} \quad C_P = K_P \quad C_A = 1 \quad k_{TA} = 0$$

Sono stati scelti i seguenti parametri:

$$K_P = 5 \quad K_V = 10 \quad T_V = T_m \quad k_{TP} = k_{TV} = 1$$

Si è imposto  $T_V = T_m$  per cancellare gli effetti del polo reale del motore in  $s = -\frac{1}{T_m}$

Per gli schemi di controllo caratterizzati dal controllo di posizione, velocità e accelerazione lo l'azione di controllo sarà caratterizzato dai controllori  $P$  di posizione e di velocità e dal controllore  $PI$  di accelerazione:

$$C_A(s) = K_A \frac{(1 + sT_A)}{s} \quad C_P = K_P \quad C_V = K_V$$

sono stati scelti i seguenti parametri:

$$K_P = 5 \quad K_V = 10 \quad K_A = 2 \quad T_A = T_m \quad k_{TP} = k_{TV} = k_{TA} = 1$$

Si è imposto  $T_A = T_m$  per cancellare gli effetti del polo reale del motore in  $s = -\frac{1}{T_m}$

Come traiettoria di riferimento  $\vartheta_r$  per entrambi i giunti è stata utilizzata la traiettoria a profilo trapezoidale di velocità, descritta in precedenza nel paragrafo dedicato.

I blocchi che rappresentano la parte non lineare del sistema, quindi gli effetti di accoppiamento tra i giunti, dovuti alla configurazione e al moto del manipolatore, sono stati implementati come segue:

$$\begin{aligned} d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} &= \begin{bmatrix} K_{r1}^{-1} & 0 \\ 0 & K_{r2}^{-1} \end{bmatrix} \begin{bmatrix} \Delta B_{11} & \Delta B_{12} \\ \Delta B_{21} & 0 \end{bmatrix} \begin{bmatrix} K_{r1}^{-1} & 0 \\ 0 & K_{r2}^{-1} \end{bmatrix} \begin{bmatrix} \ddot{q}_{m1} \\ \ddot{q}_{m2} \end{bmatrix} \\ &+ \begin{bmatrix} K_{r1}^{-1} & 0 \\ 0 & K_{r2}^{-1} \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & 0 \end{bmatrix} \begin{bmatrix} K_{r1}^{-1} & 0 \\ 0 & K_{r2}^{-1} \end{bmatrix} \begin{bmatrix} \dot{q}_{m1} \\ \dot{q}_{m2} \end{bmatrix} + \begin{bmatrix} K_{r1}^{-1} & 0 \\ 0 & K_{r2}^{-1} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \\ &= \begin{bmatrix} K_{r1}^{-1} \Delta B_{11} K_{r1}^{-1} \ddot{q}_{m1} + K_{r1}^{-1} \Delta B_{12} K_{r2}^{-1} \ddot{q}_{m2} \\ K_{r2}^{-1} \Delta B_{21} K_{r1}^{-1} \ddot{q}_{m1} \end{bmatrix} + \begin{bmatrix} K_{r1}^{-1} c_{11} K_{r1}^{-1} \dot{q}_{m1} + K_{r1}^{-1} c_{12} K_{r2}^{-1} \dot{q}_{m2} \\ K_{r2}^{-1} c_{21} K_{r1}^{-1} \dot{q}_{m1} \end{bmatrix} \\ &+ \begin{bmatrix} K_{r1}^{-1} g_1 \\ K_{r2}^{-1} g_2 \end{bmatrix} => \end{aligned}$$

$$\Rightarrow d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} K_{r1}^{-1} \Delta B_{11} K_{r1}^{-1} \ddot{q}_{m1} + K_{r1}^{-1} \Delta B_{12} K_{r2}^{-1} \ddot{q}_{m2} + K_{r1}^{-1} c_{11} K_{r1}^{-1} \dot{q}_{m1} + K_{r1}^{-1} c_{12} K_{r2}^{-1} \dot{q}_{m2} + K_{r1}^{-1} g_1 \\ K_{r2}^{-1} \Delta B_{21} K_{r1}^{-1} \ddot{q}_{m1} + K_{r2}^{-1} c_{21} K_{r1}^{-1} \dot{q}_{m1} + K_{r2}^{-1} g_2 \end{bmatrix}$$

Dove

$$\Delta B_{11} = 2m_2 L_1 d_2 \mathbf{c}_2 \quad \Delta B_{12} = \Delta B_{21} = m_2 L_1 d_2 \mathbf{c}_2$$

$$c_{11} = -m_2 L_1 d_2 s_2 \dot{\vartheta}_2 = h s_2 \dot{\vartheta}_2 \quad c_{12} = -m_2 L_1 d_2 s_2 (\dot{\vartheta}_1 + \dot{\vartheta}_2) = h s_2 (\dot{\vartheta}_1 + \dot{\vartheta}_2)$$

$$c_{21} = m_2 L_1 d_2 s_2 \dot{\vartheta}_1 = -h s_2 \dot{\vartheta}_1$$

$$\textcolor{red}{g_1} = (m_1 d_1 + m_2 L_1 + m_{m2} L_1) * g c_1 + m_2 d_2 g c_{12} = g_{1,1} \mathbf{c}_1 + g_{1,2} \mathbf{c}_{12} \quad \textcolor{red}{g_2} = m_2 d_2 g c_{12} = g_2 \mathbf{c}_{12}$$

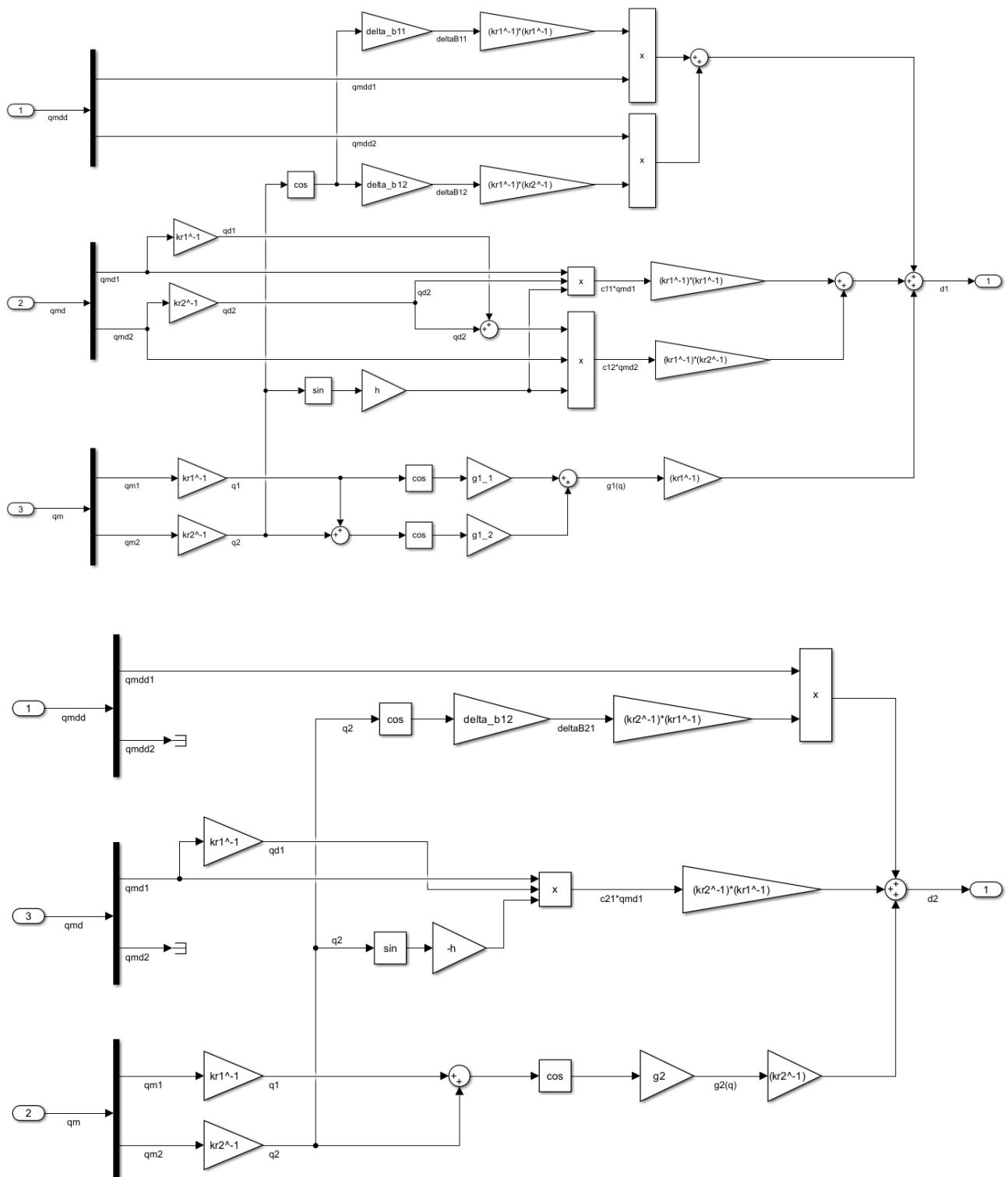


Figura 14: Implementazione dei disturbi  $d1$  e  $d2$  in Simulink

Tutti i tipi di controllo indipendente ai giunti saranno caratterizzati da questi blocchi:

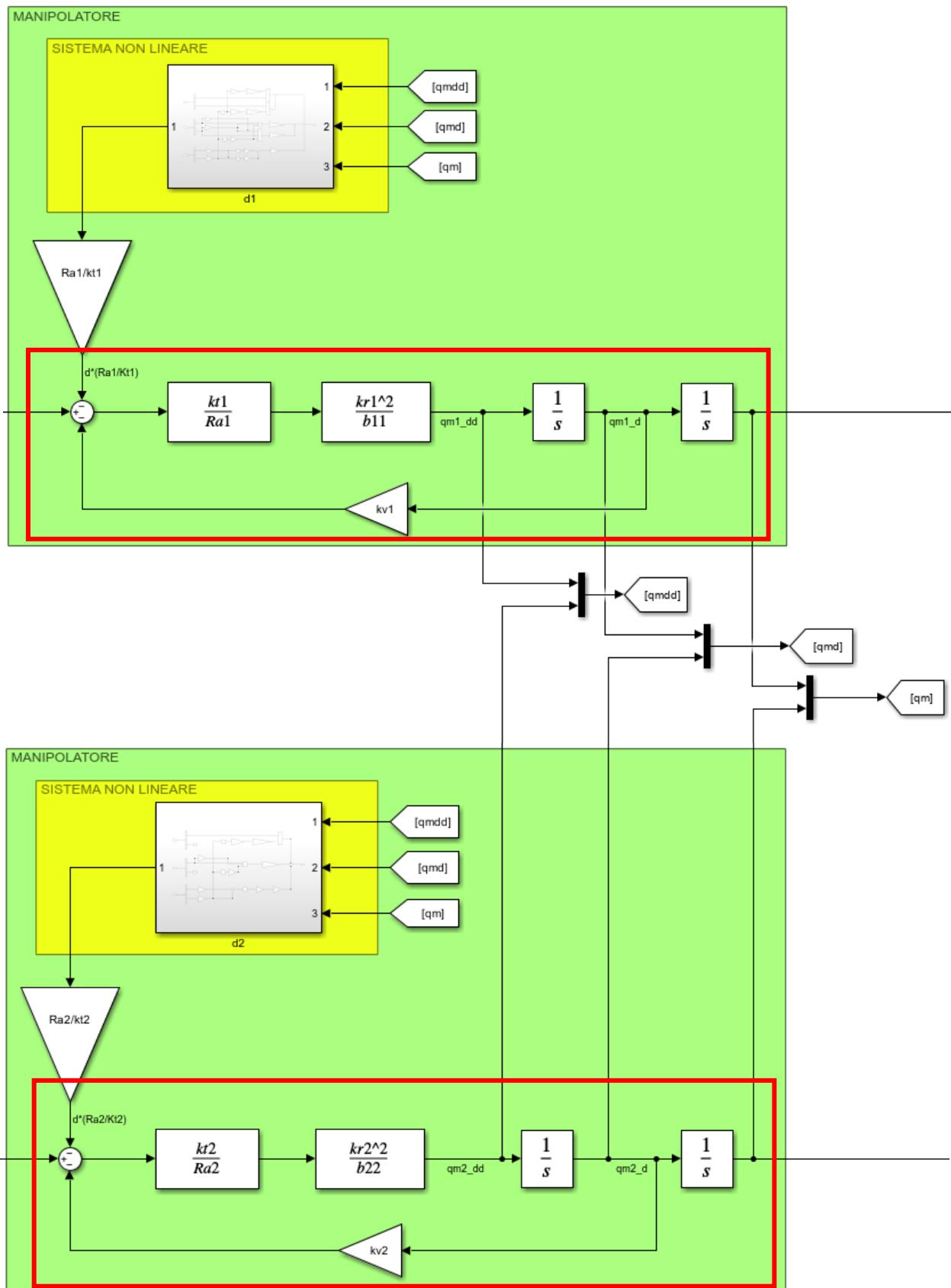


Figura 15: Implementazione generale del controllo indipendente ai giunti in Simulink

Dove l'area in giallo rappresenta il sistema non lineare, infatti comprende il modello dinamico del manipolatore, utilizzato per valutare il disturbo  $d$ .

Invece, l'area in verde può essere vista come il manipolatore, in quanto in ingresso c'è la coppia.

Infine, i blocchi evidenziati rappresentano motore  $M(s)$ , quindi lo schema di controllo degli electric drives che “guidano” i giunti.

### 5.1.2 Implementazione Feedback control

Il controllo in retroazione è il più semplice tipo di controllo che si può realizzare.

Il position feedback control è stato implementato nel seguente modo:

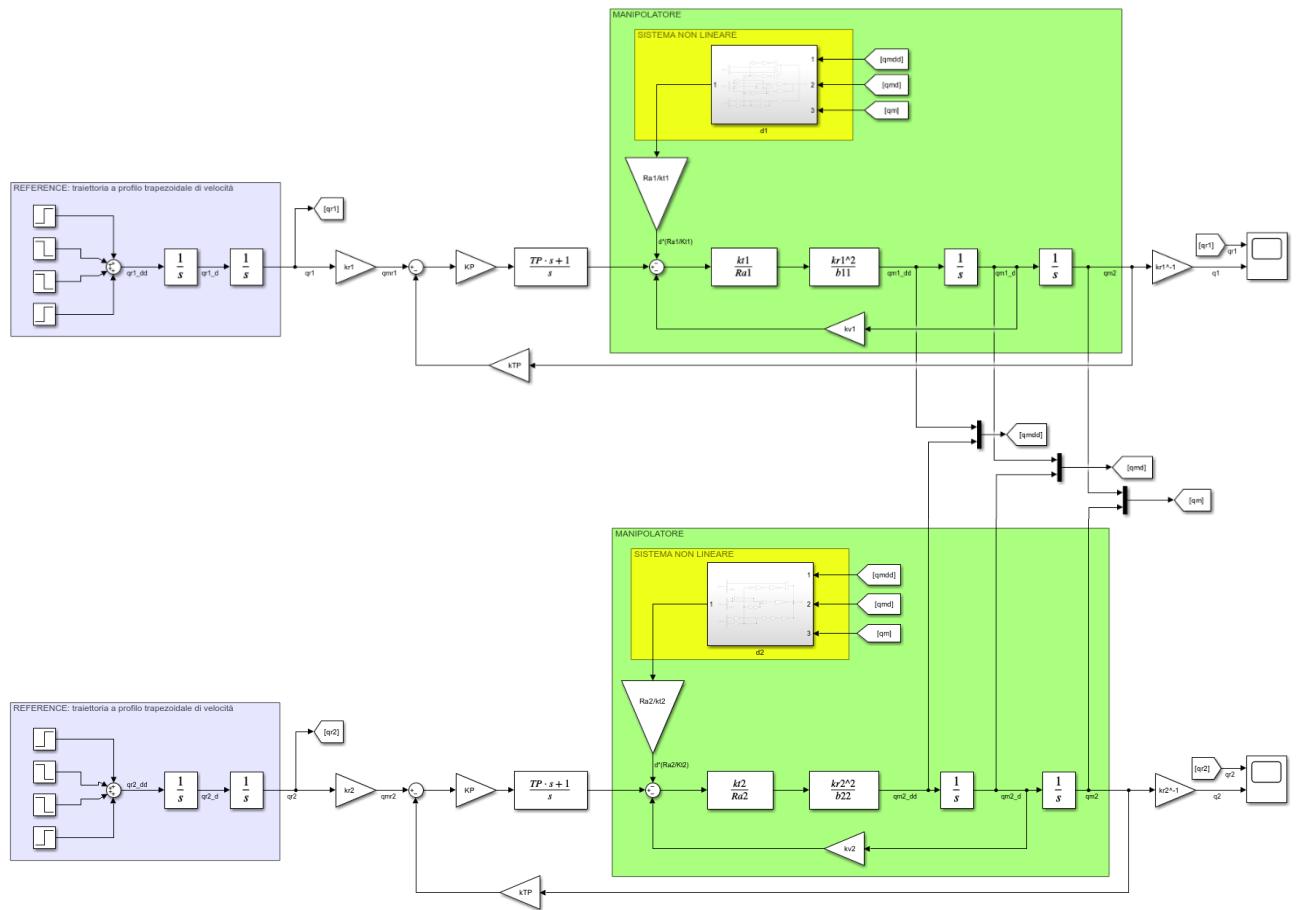
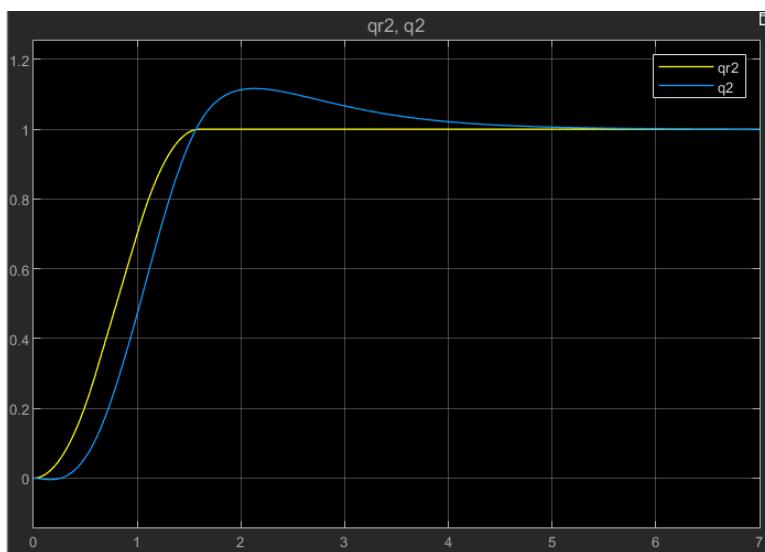
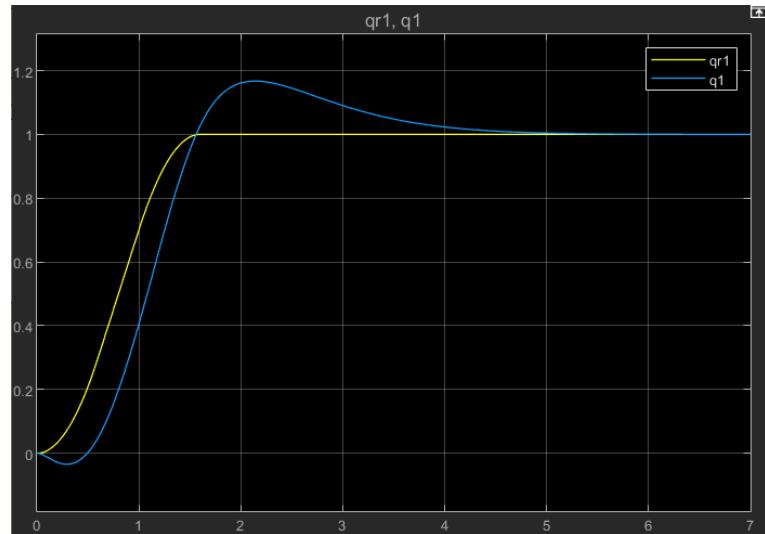


Figura 16: Implementazione Simulink del feedback position control

La simulazione mostra come questo tipo di controllo abbia una prestazione scadente:



*Figura 17: Simulazione feedback position control*

Si nota un elevato errore in quanto la traiettoria che effettivamente seguono le due variabili di giunto è diversa da quella di riferimento. Soprattutto la prima variabile di giunto  $q_1$ , che risente di più l'inerzia del sistema, presenta un sottosmorzamento nei primi 1.5 secondi, per poi avere un picco verso i 2 secondi.

Le variabili si assestano solo dopo 5.8 secondi, quindi ben oltre la fine della traiettoria al tempo  $t = 2s$ .

Il controllo in retroazione di posizione e velocità è stato implementato nel seguente modo:

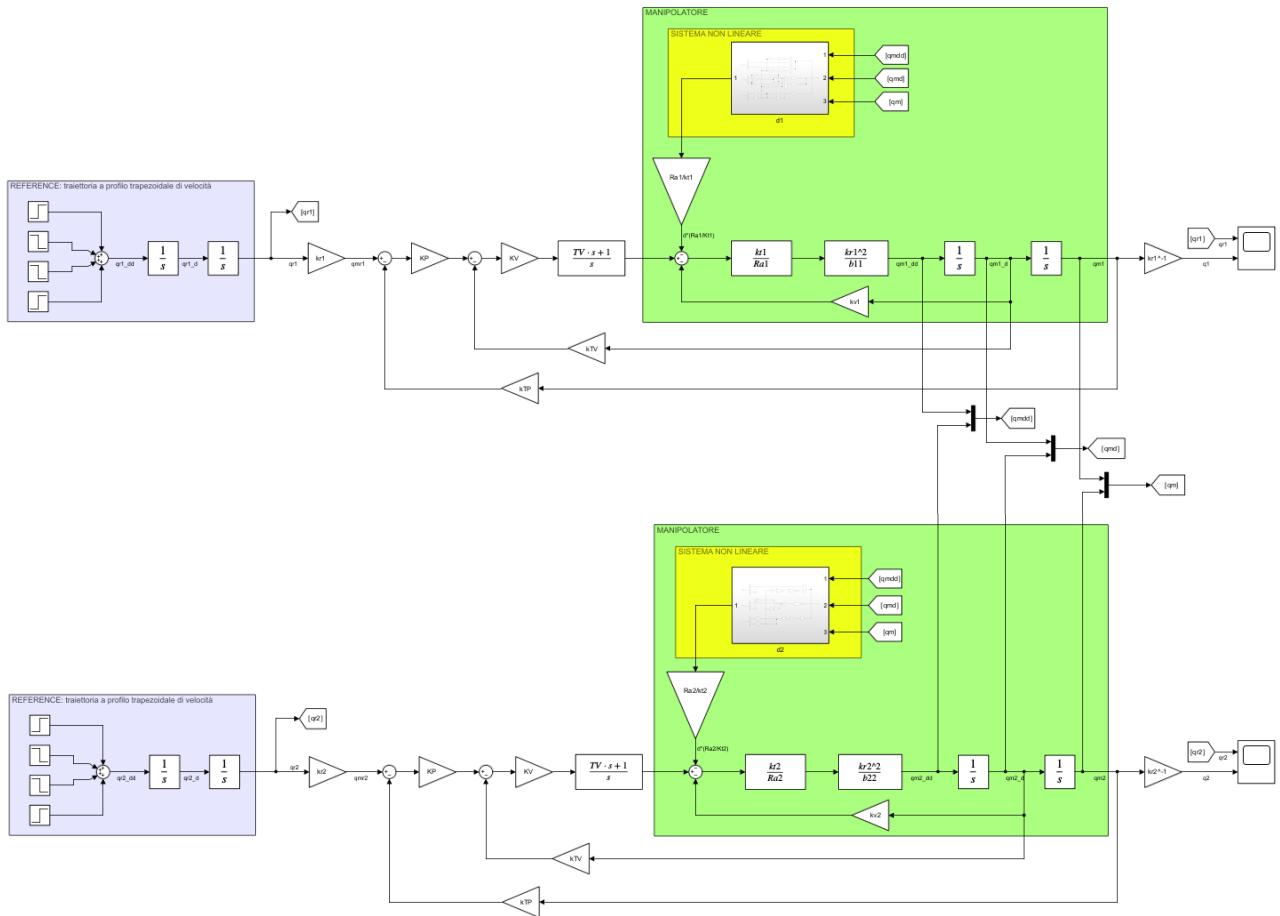
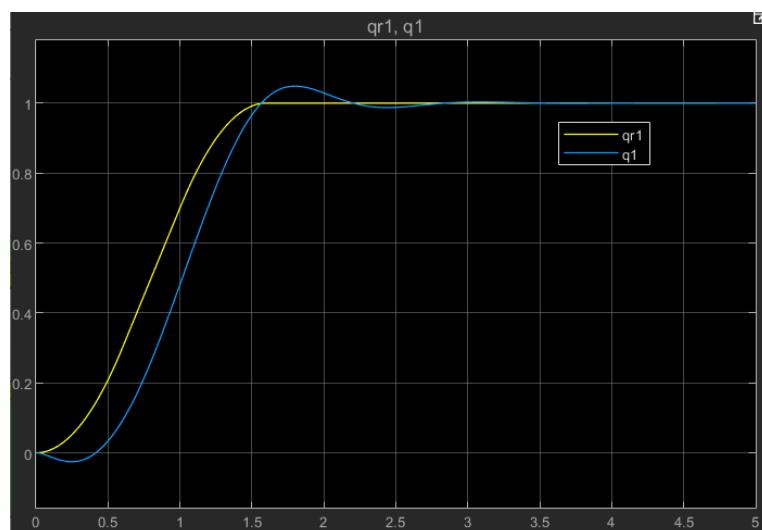
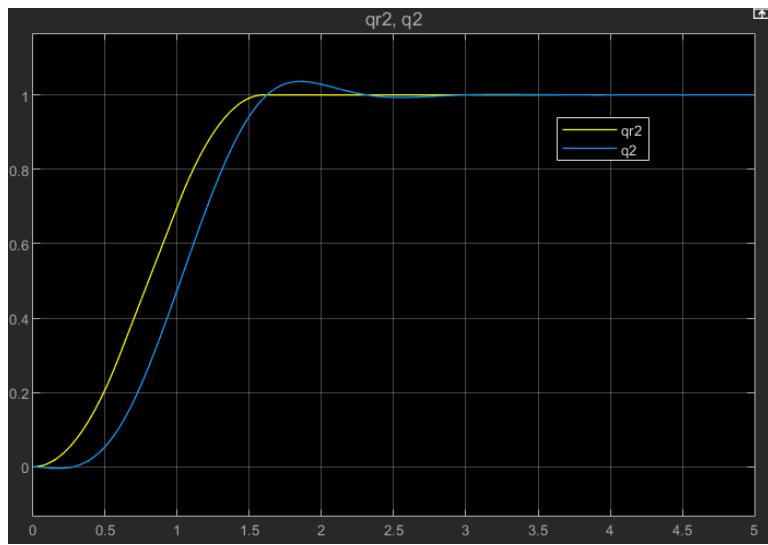


Figura 18: Implementazione Simulink del feedback position and velocity control

La simulazione è la seguente:





*Figura 19: Simulazione feedback position and velocity control*

La simulazione mostra come, con la sola aggiunta della retroazione di velocità, il valore del picco è diminuito. Comunque, è ancora presente un elevato errore per via della distanza dalla traiettoria di riferimento.

Il controllo in retroazione che comprende tutte le tre retroazioni, di posizione velocità e accelerazione, è stato implementato nel seguente modo:

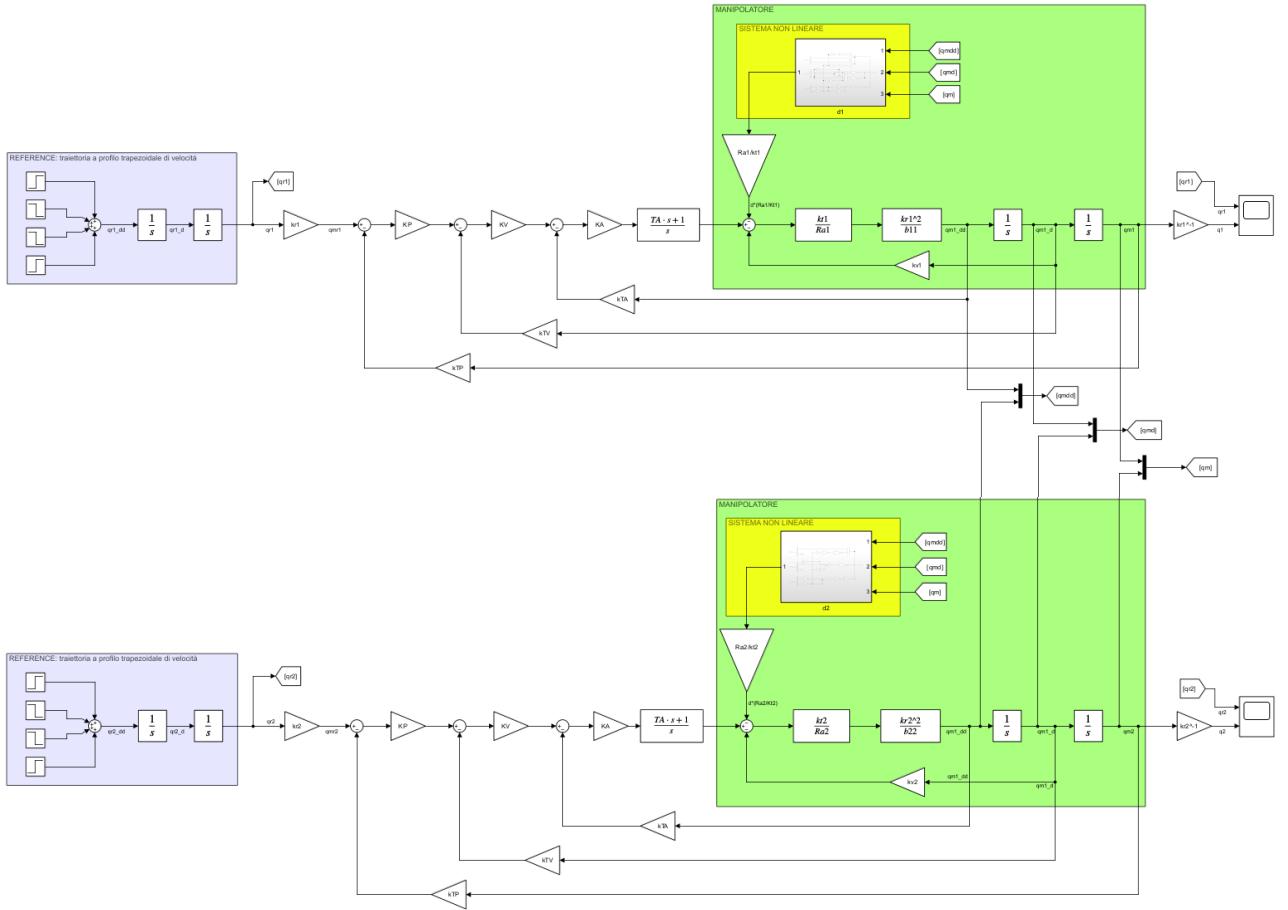
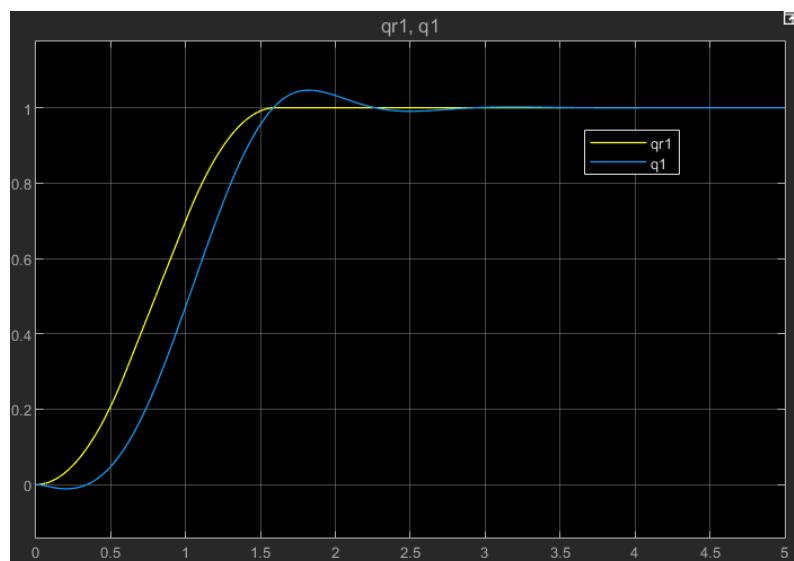
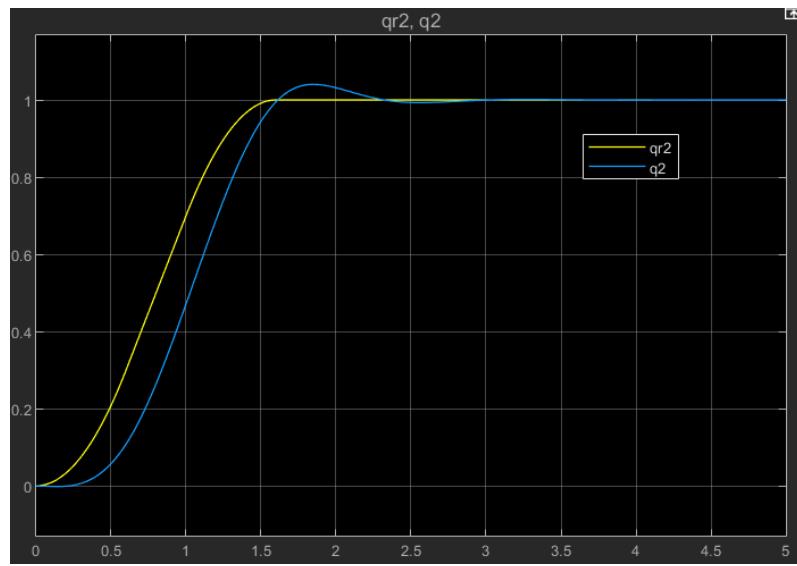


Figura 20: Implementazione Simulink del feedback position, velocity and acceleration control

La simulazione è pressoché simile a quella precedente:





*Figura 21: Simulazione feedback position, velocity and acceleration control*

### 5.1.3 Compensazione in avanti decentralizzata

Avendo considerato una traiettoria di riferimento con elevati valori di velocità e accelerazioni (traiettoria veloce), le capacità di inseguimento degli schemi di controllo sopra citati sono limitate.

L'impiego di un'azione di *compensazione in avanti decentralizzata* consente di ridurre l'errore di inseguimento e quindi migliorare la performance di tracking.

L'azione in avanti consiste nell'introduzione di due termini che sono proporzionali ai riferimenti di velocità e accelerazione.

Per ognuno dei tre casi presentati nel paragrafo precedente possono essere individuati due diversi schemi di controllo, il primo che considera solo la compensazione in avanti, il secondo che rappresenta uno schema equivalente. Quest'ultimo, infatti, invece di adottare delle retroazioni di variabili intermedie, utilizza la sola retroazione di posizione e regolatori con azioni standard, rispettivamente per i tre casi presentati, un *PI*, un *PID* e un *PIDD*<sup>2</sup>.

Le soluzioni sono equivalenti dal punto di vista della reiezione dei disturbi e dell'inseguimento della traiettoria; tuttavia, l'individuazione dei pesi per l'azione di controllo è meno immediata. Quindi si è deciso di implementare solo il primo schema di controllo per ognuno dei tre casi sopra citati.

#### *Implementazione*

Per il caso di sola retroazione di posizione introduciamo i seguenti termini:

$$\frac{T_m}{k_m} \quad \frac{1}{k_m}$$

È ovviamente richiesta la conoscenza della costante di tempo  $T_m$  e del guadagno di tensione  $k_m$  ( $= \frac{1}{k_v}$ ) di motore.

L'implementazione è la seguente:

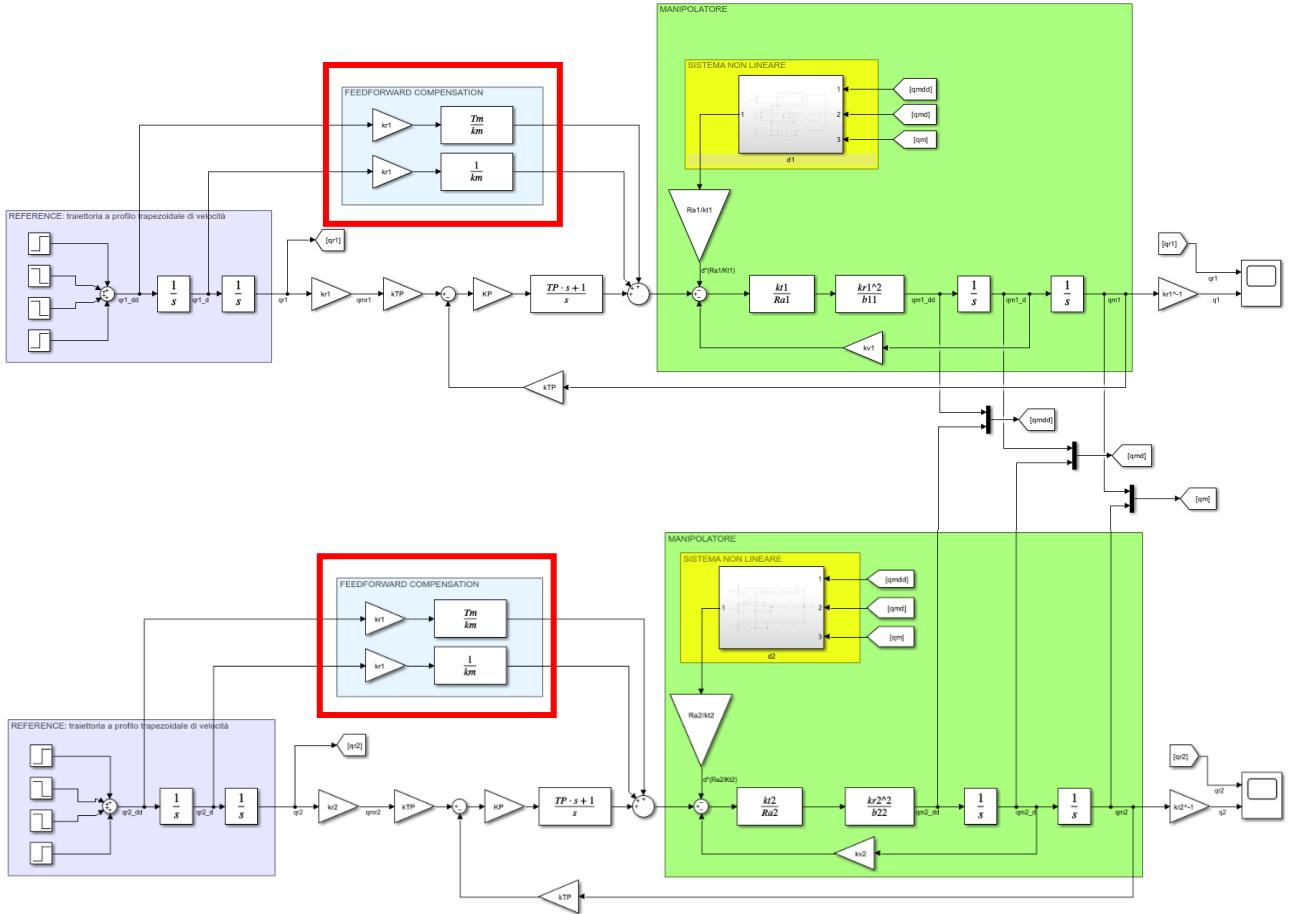
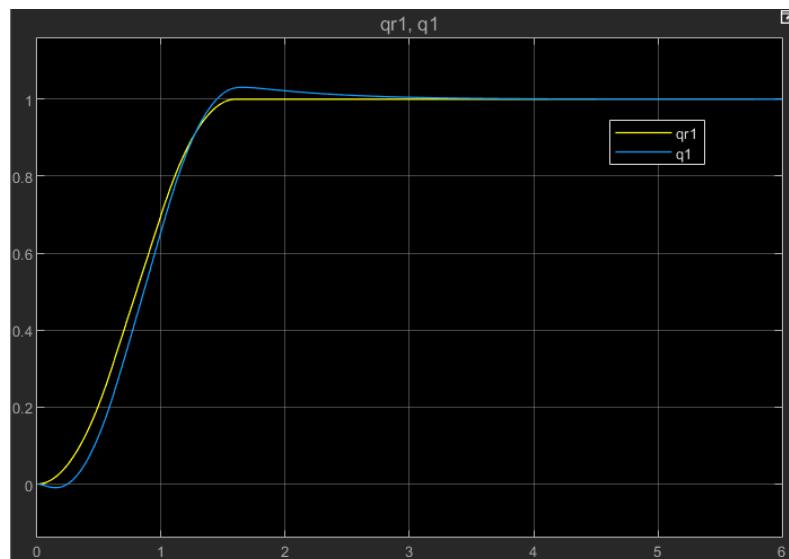
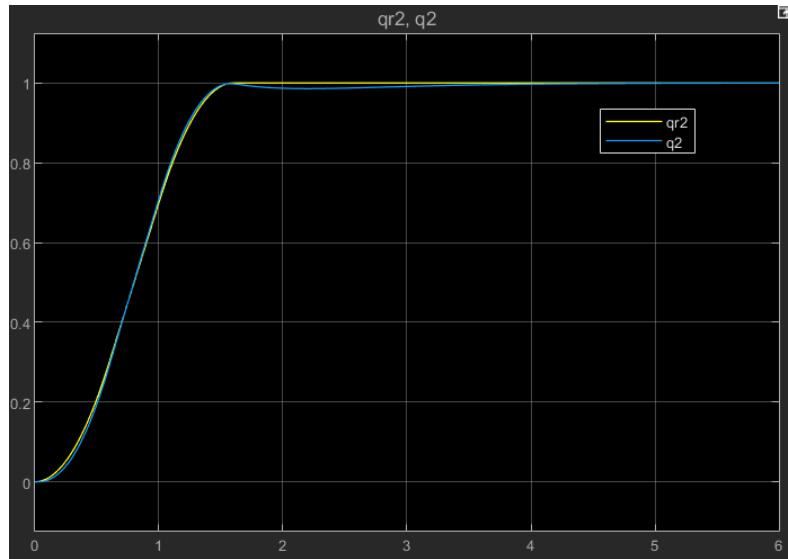


Figura 22: Implementazione con compensazione in avanti decentralizzata - posizione

Le parti evidenziate in rosso sono quelle aggiunte rispetto alla configurazione senza compensazione in avanti.

La simulazione è la seguente:





*Figura 23: Simulazione compensazione in avanti decentralizzata - posizione*

La simulazione mostra un netto miglioramento, soprattutto per la seconda variabile, rispetto al caso di sola retroazione:

- La prima variabile di giunto presenta ancora un leggero sovrasmorzamento e una distanza dalla traiettoria di riferimento.
- La seconda variabile invece presenta un buon inseguimento della traiettoria e un leggero sottosmorzamento prima di assestarsi al valore finale.

Comunque, si è ancora lontani da una buona performance di tracking.

Per il caso di retroazione di posizione e velocità introduciamo i seguenti termini:

$$\frac{1}{k_m K_V} \quad k_{TV}$$

In questo caso si richiede la conoscenza della costante di trasduzione  $k_{TV}$ , della costante proporzionale  $K_V$  e di  $k_m$ .

L'implementazione è la seguente, dove le parti evidenziate in rosso sono quelle aggiunte rispetto alla configurazione senza compensazione in avanti:

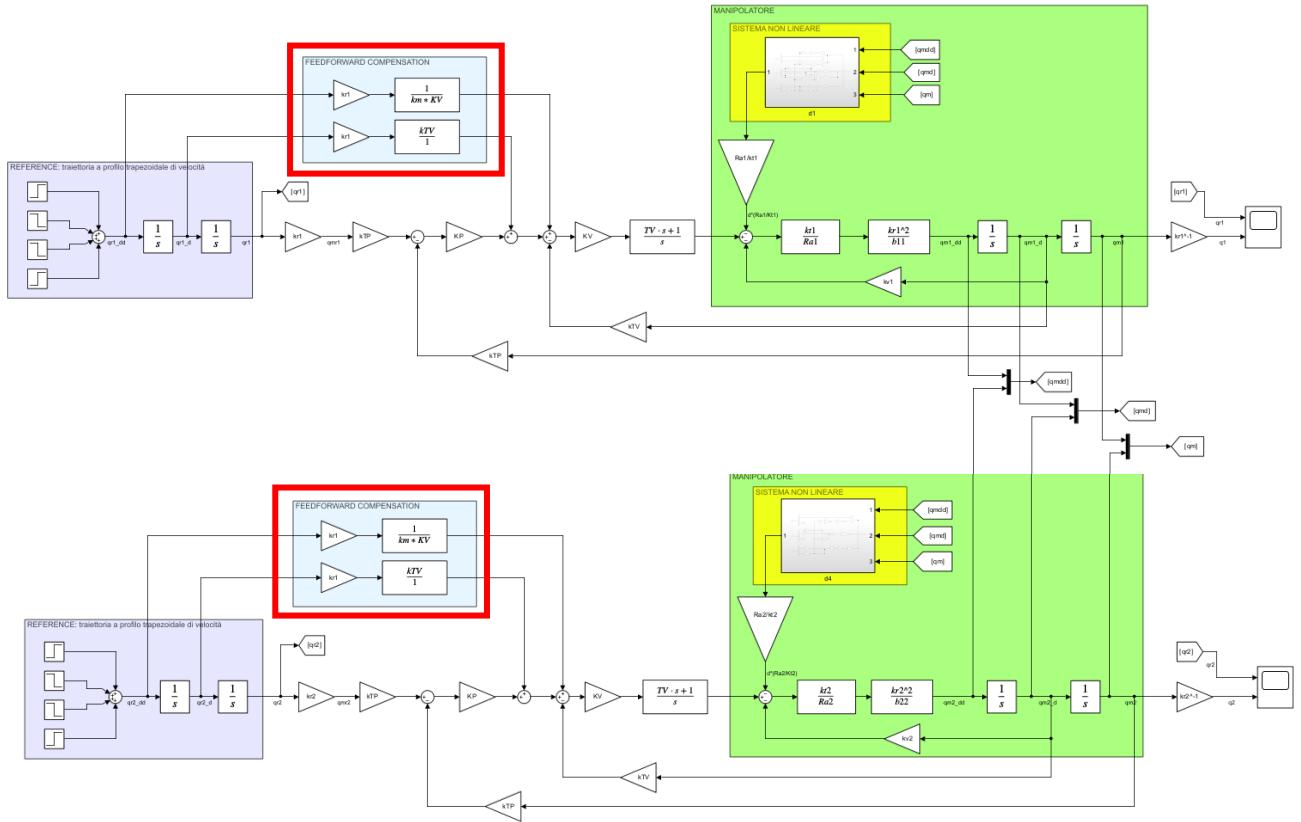
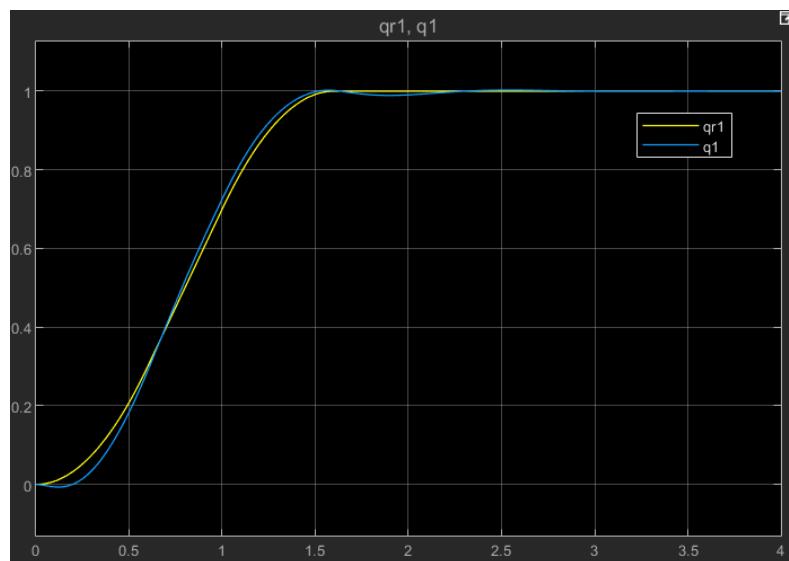
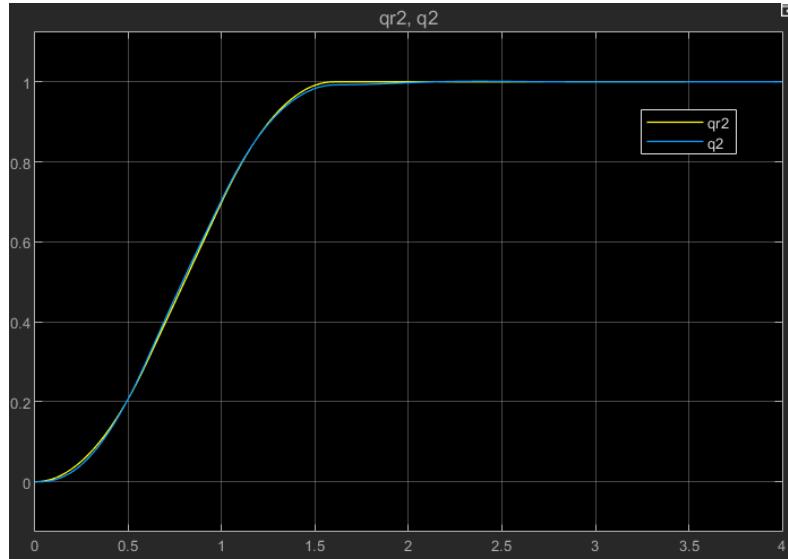


Figura 24: Implementazione con compensazione in avanti decentralizzata – posizione e velocità

La simulazione è la seguente:





*Figura 25: Simulazione compensazione in avanti decentralizzata - posizione e velocità*

Si nota ancora un miglioramento delle prestazioni. L'errore diminuisce sia per la prima che per la seconda variabile. La prima ha comunque ancora problemi a seguire il riferimento e ad assestarsi prima della conclusione della traiettoria.

La seconda presenta una buona un minimo errore con ancora un minimo sottosmorzamento prima di assestarsi definitivamente.

Per il caso di retroazione di posizione, velocità e accelerazione introduciamo i seguenti termini:

$$k_{TV} = k_{TA} + \frac{1}{k_m K_A}$$

In questo caso si richiede la conoscenza delle costanti di trasduzione  $k_{TV}$  e  $k_{TA}$ , della costante proporzionale  $K_A$  e di  $k_m$ .

L'implementazione è la seguente:

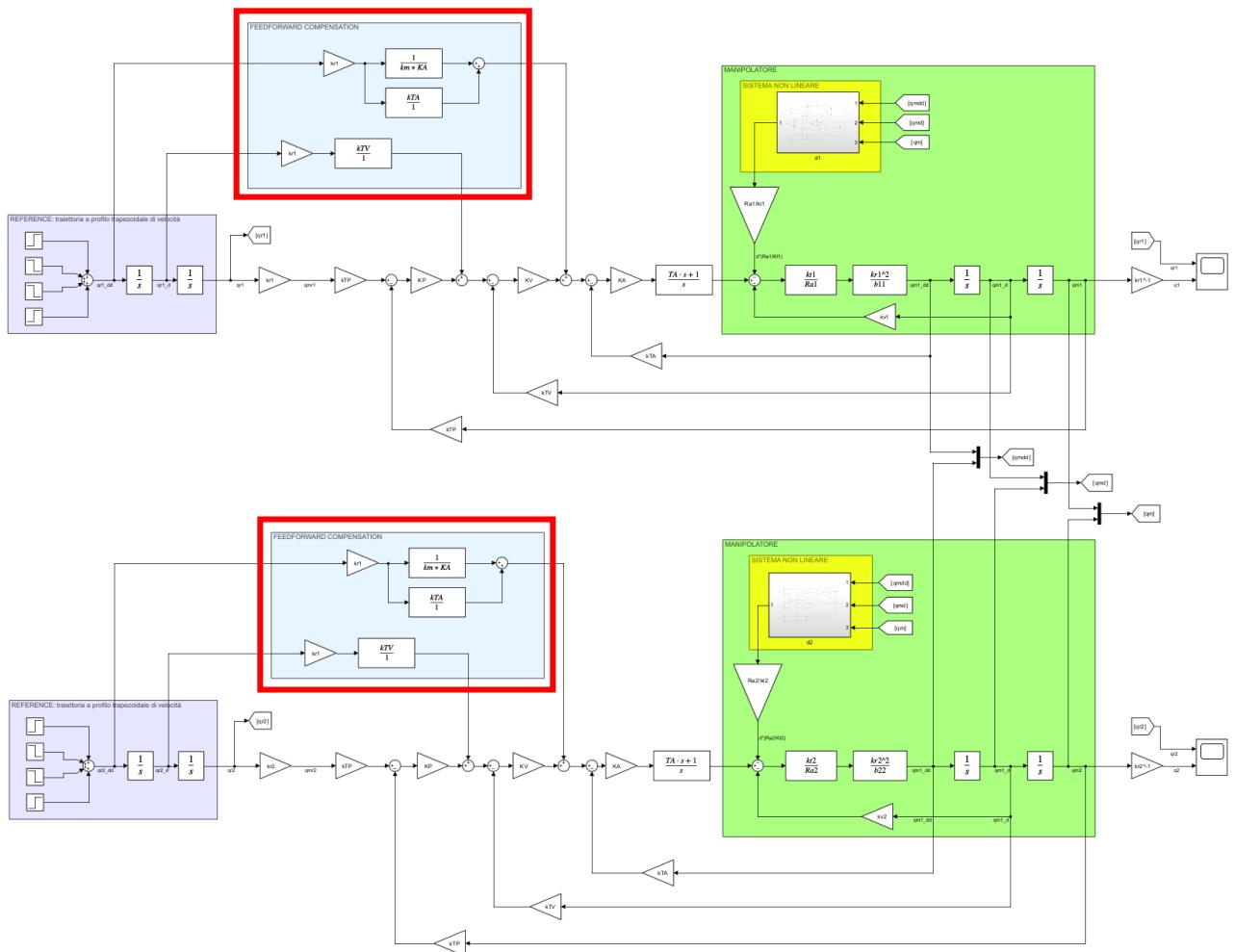
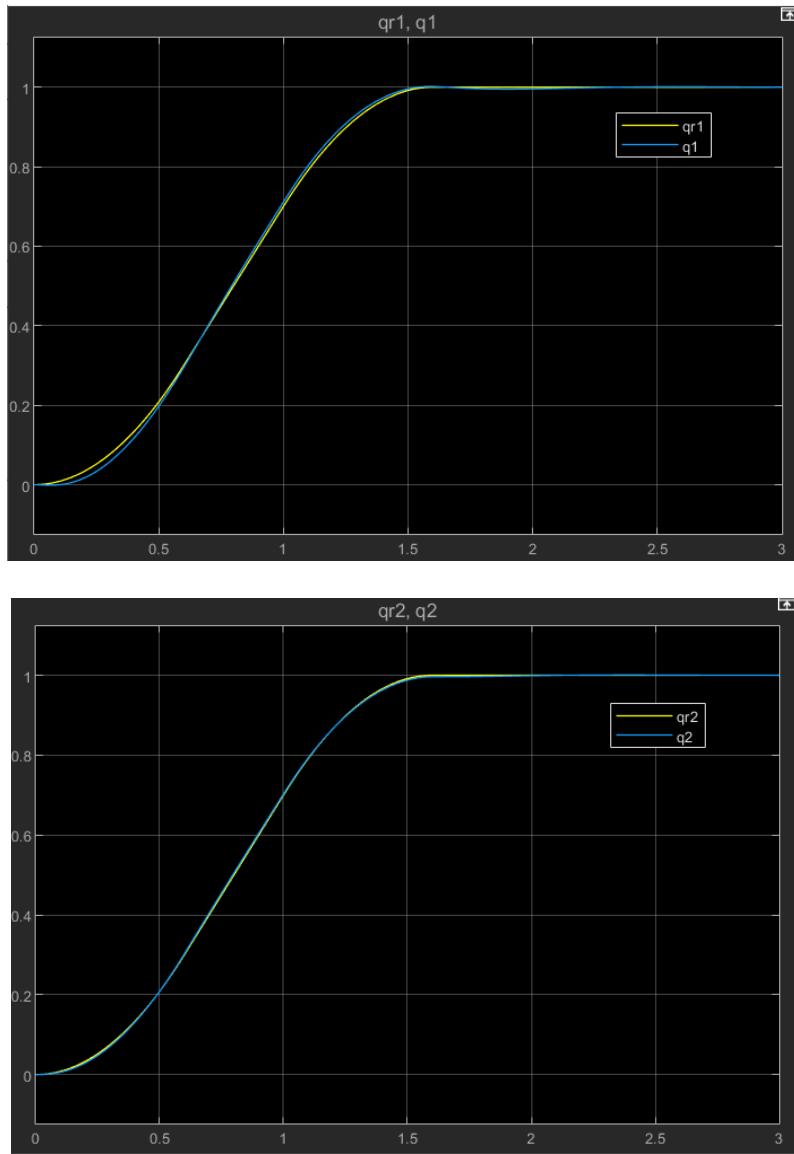


Figura 26: Implementazione con compensazione in avanti decentralizzata – posizione, velocità e accelerazione

Le parti evidenziate in rosso sono quelle aggiunte rispetto alla configurazione senza compensazione in avanti.



*Figura 27: Simulazione compensazione in avanti decentralizzata - posizione, velocità e accelerazione*

Con quest'ultima simulazione si può notare un netto miglioramento delle prestazioni di tracking della traiettoria.

La prima variabile presenta ancora delle minime oscillazioni attorno al riferimento; invece, la seconda variabile presenta un quasi perfetto inseguimento della traiettoria.

Si può notare come l'errore sia minimo per entrambe le variabili di giunto:

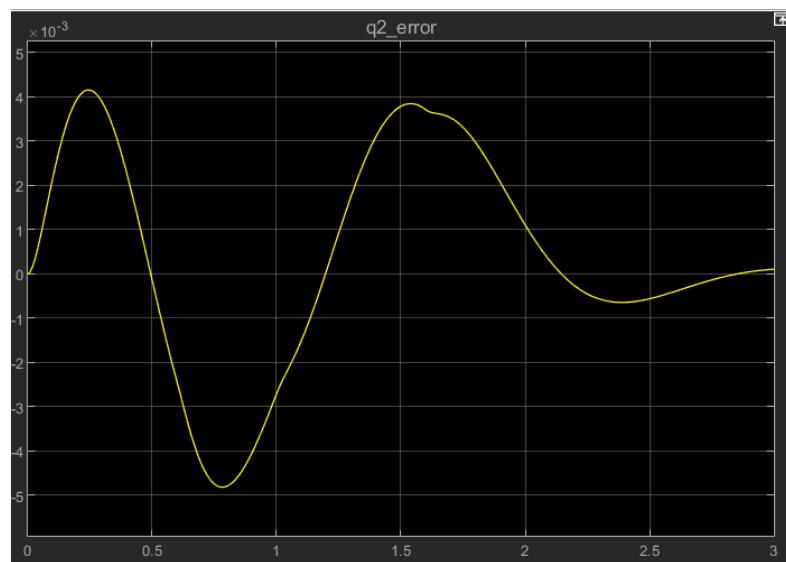
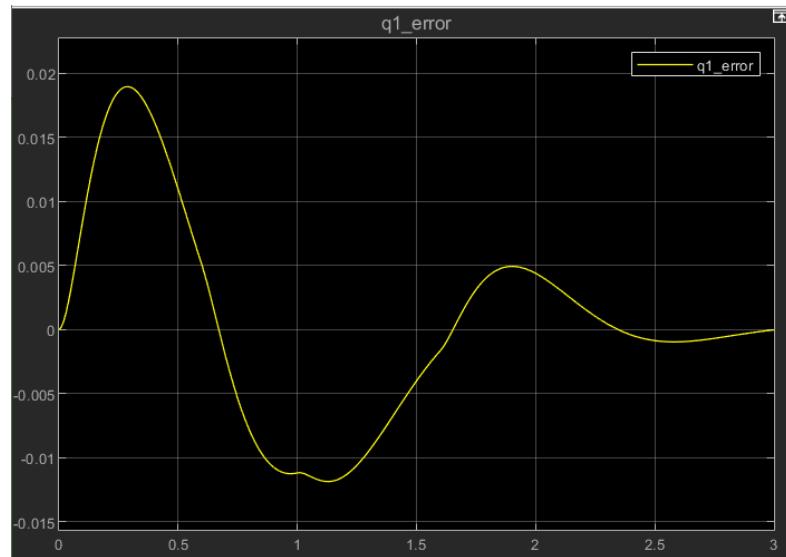


Figura 28: Errore compensazione in avanti decentralizzata – posizione, velocità e accelerazione

L'errore per la seconda variabile risulta di un ordine di grandezza pari a  $10^{-3}$ .

### 5.1.4 Compensazione in avanti a coppia precalcolata

In presenza di un disturbo  $d(t) = 0$ , ogni traiettoria è asintoticamente seguita. Infatti, una presenza del disturbo comporta la nascita di un errore di inseguimento.

Anche se  $d(t)$  è stato considerato come disturbo, di esso se ne conosce l'espressione, quindi si può aggiungere ai segnali di azione in avanti un termine che non compensa gli effetti, ma il disturbo stesso.

Si aggiunge quindi il seguente termine:

$$R_a k_t^{-1} d_r$$

Con  $R_a$  matrice diagonale delle resistenze di armatura e  $k_t$  matrice diagonale delle costanti di coppia degli attuatori.

Il termine  $d_r$  ha invece espressione:

$$d_r = k_r^{-1} \Delta B(q_d) k_r^{-1} \ddot{q}_{md} + k_r^{-1} C(q_d, \dot{q}_d) k_r^{-1} \dot{q}_{md} + k_r^{-1} g(q_d)$$

Dove  $q_d(t)$  è la traiettoria desiderata per i giunti e  $q_{md}(t)$  la corrispondente traiettoria per gli attuatori secondo la seguente espressione:

$$k_r q = q_m$$

Lo schema da implementare è il seguente:

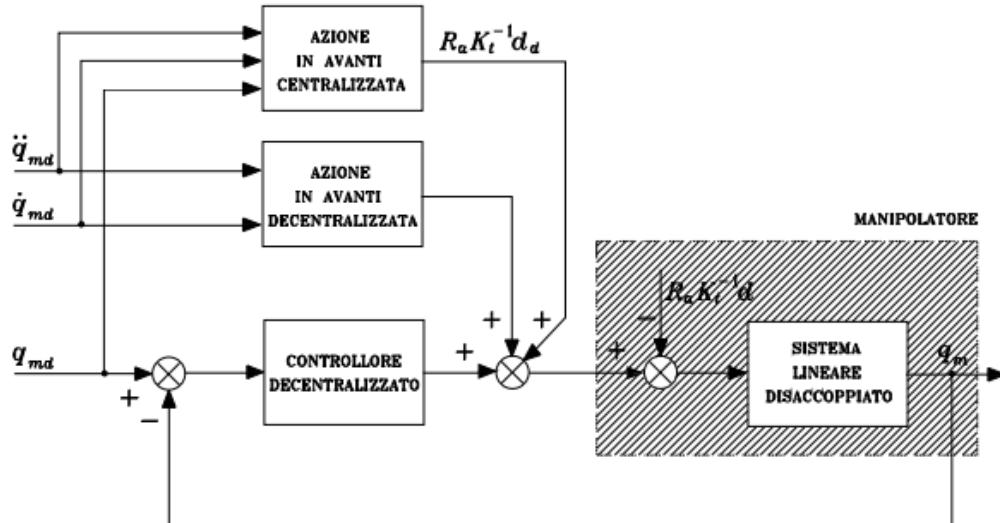


Figura 29: Schema di controllo a coppia precalcolata

## Implementazione

Di seguito sono presentate le tre implementazioni del controllo a coppia precalcolata nei tre casi di feedback control (posizione, posizione e velocità, posizione velocità e accelerazione).

Il blocco aggiuntivo è lo stesso per i tre casi di retroazione:

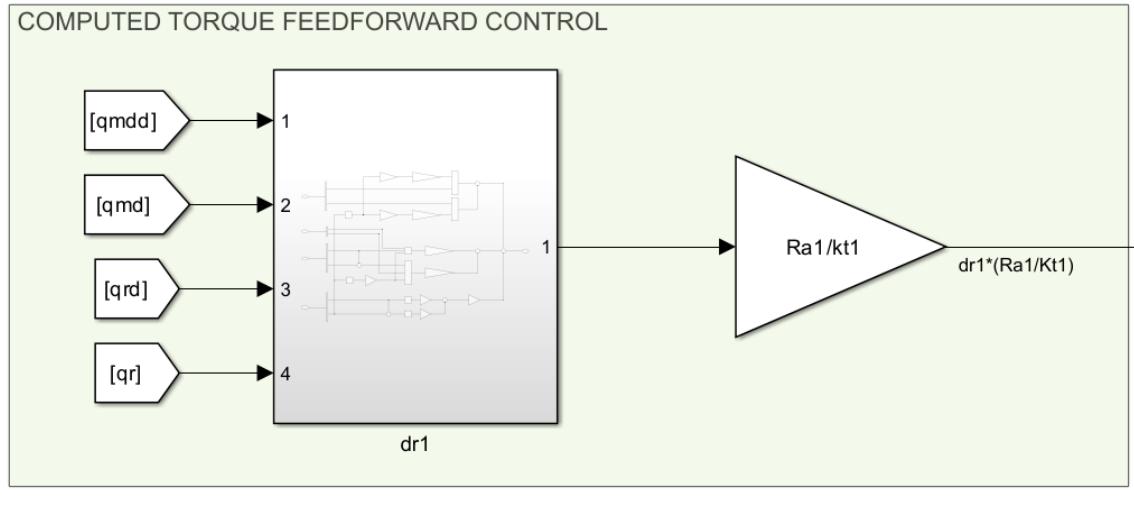


Figura 30: Blocco per implementazione coppia precalcolata

Dove  $d_{r1}$  è stato implementato come segue ( $d_{r2}$  è analogo):

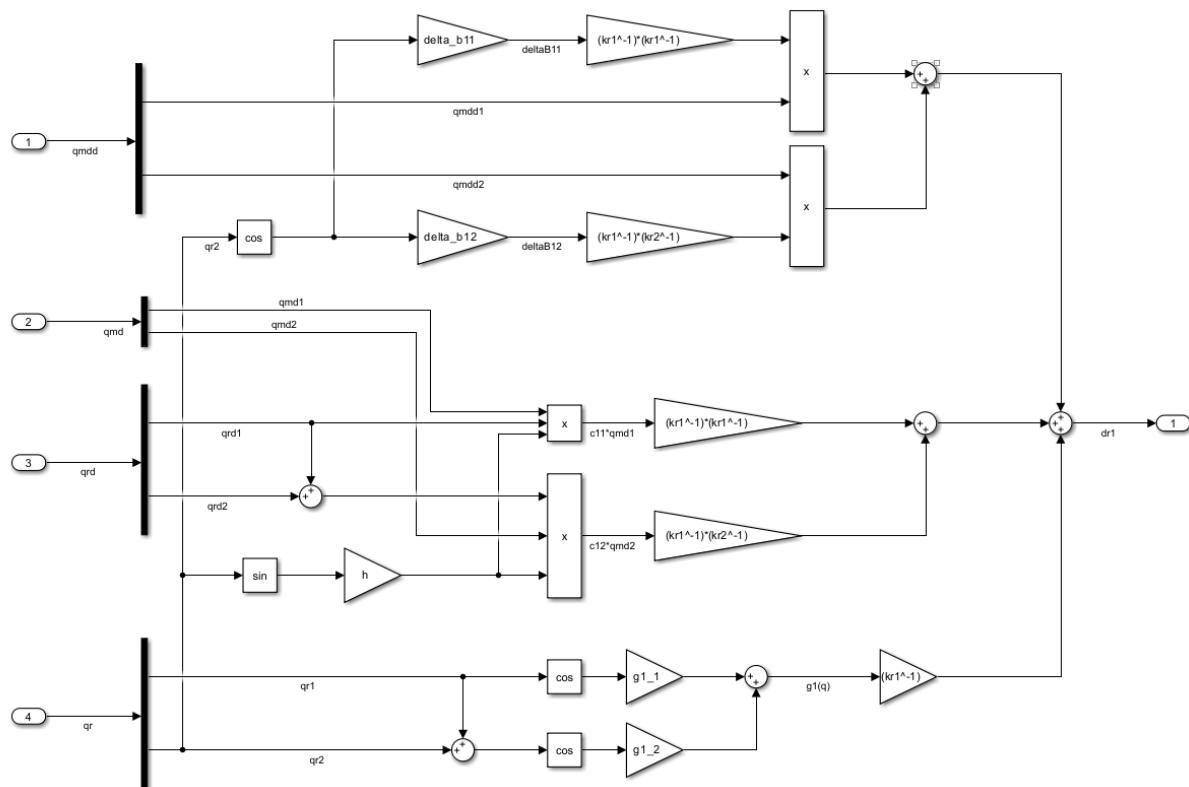


Figura 31: Implementazione blocco dr1

In seguito, sono stati evidenziati in rosso i blocchi aggiunti in questa configurazione.

Per la sola retroazione di posizione si ha:

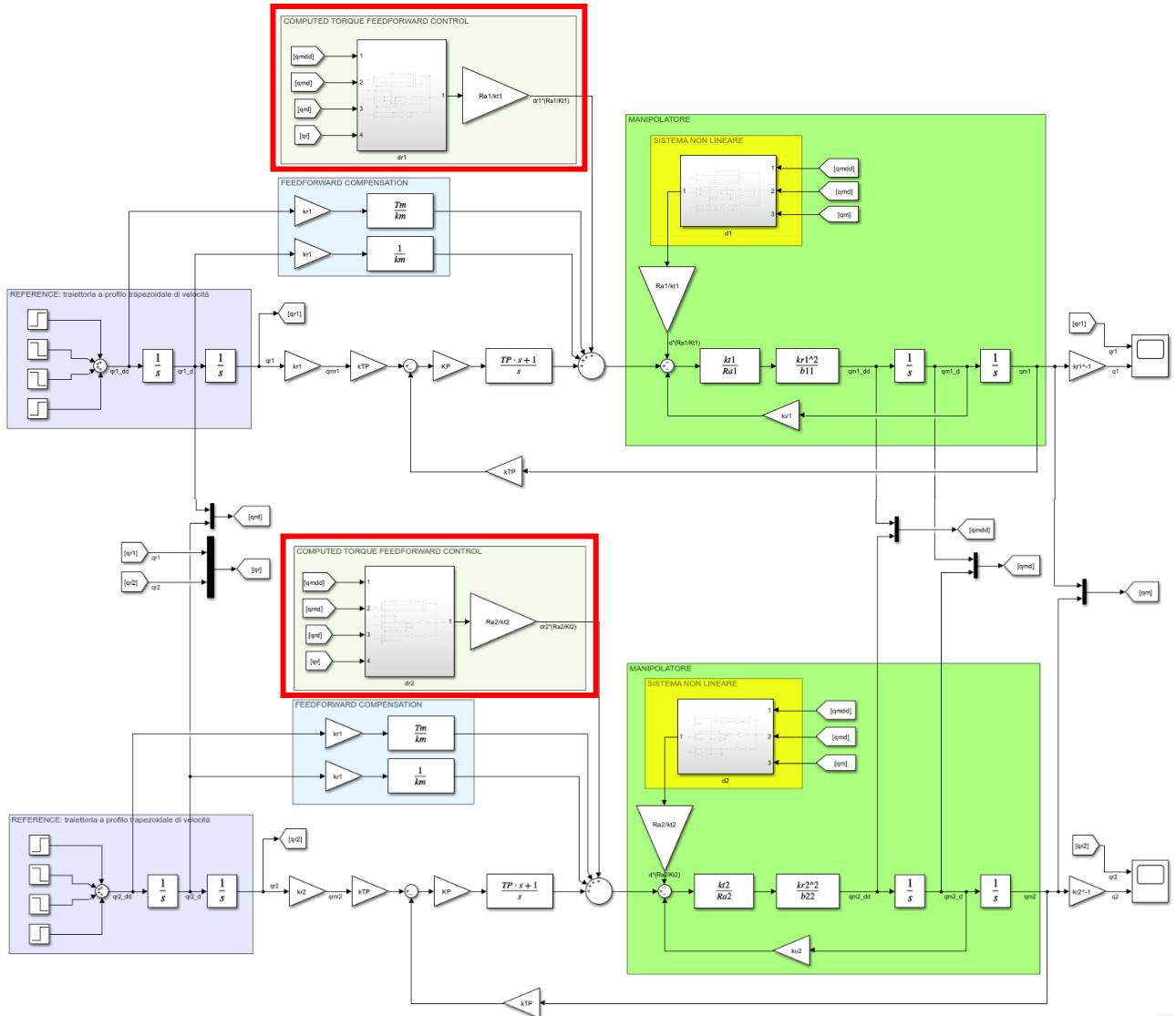


Figura 32: Implementazione Simulink coppia precalcolata - posizione

La simulazione è la seguente:

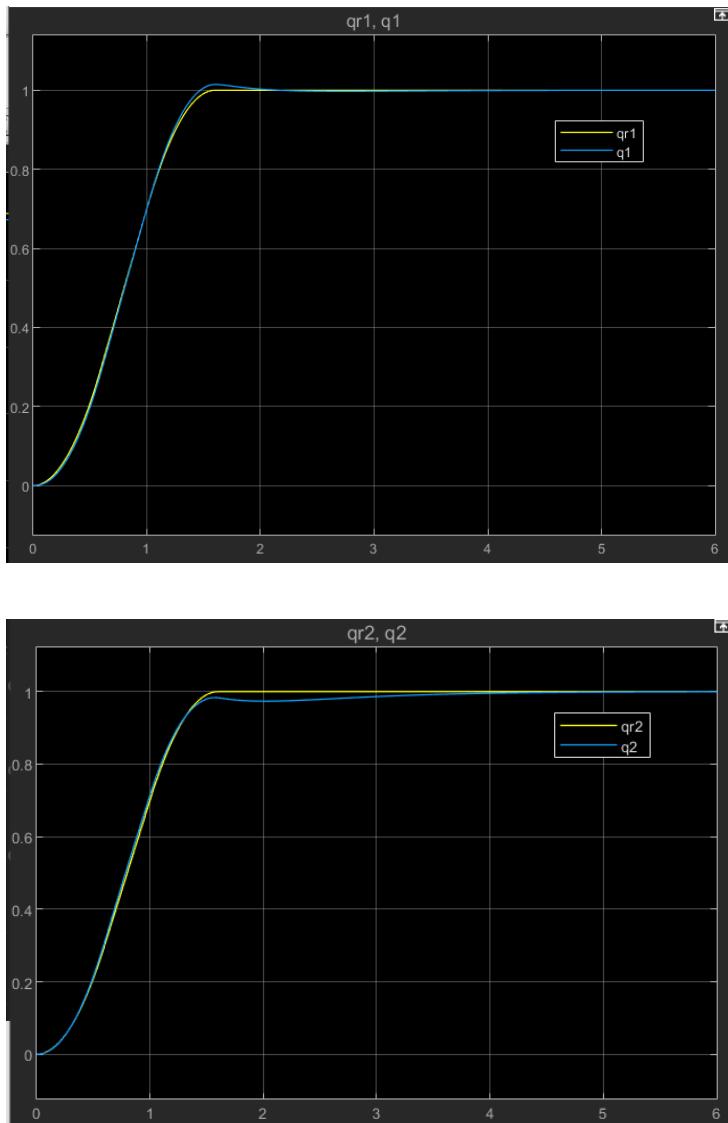


Figura 33: Simulazione coppia precalcolata - posizione

Si nota che per la prima variabile si ha un miglioramento nell'inseguimento della traiettoria, ma è sempre presente un leggero sovrasmorzamento.

La seconda variabile invece, anche se è caratterizzata da un buon inseguimento della traiettoria, presenta un eccessivo sottosmorzamento che ritarda l'assestamento alla posizione finale. Questo può essere causato da un controllo troppo aggressivo.

L'implementazione a coppia precalcolata con retroazione di posizione e velocità è invece:

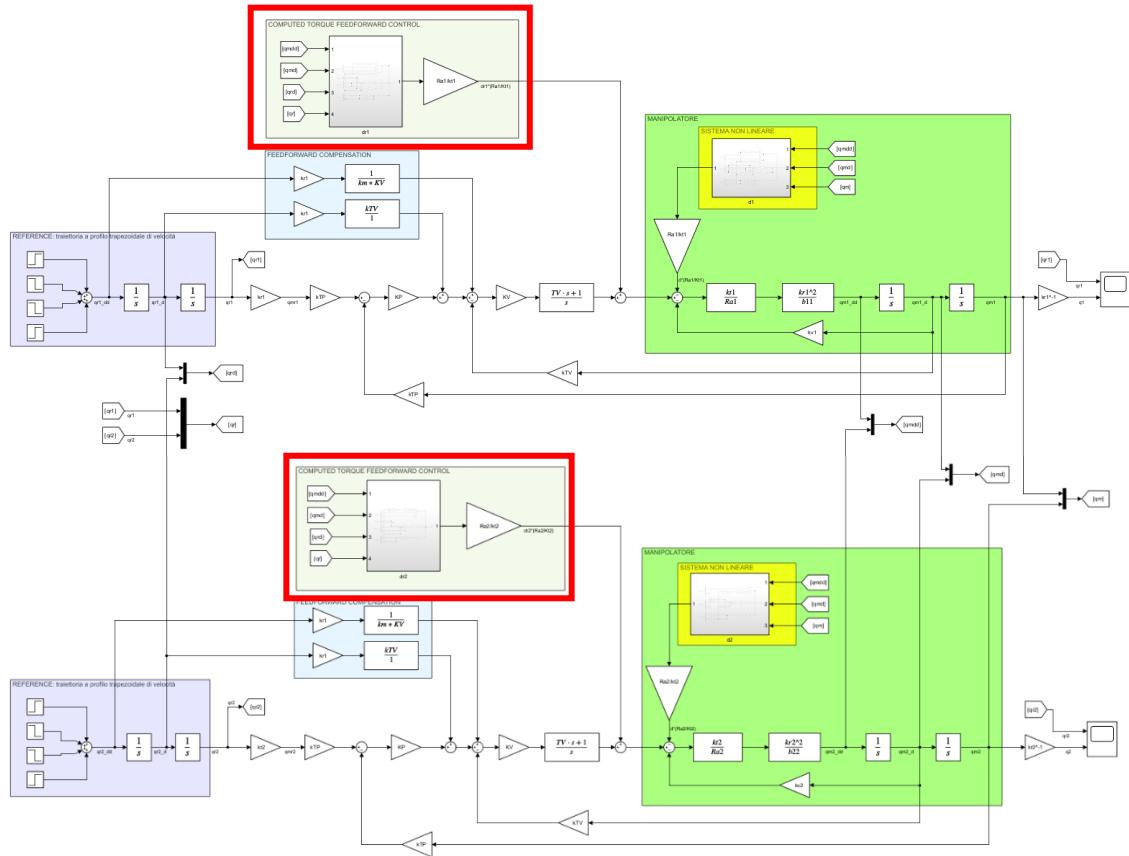
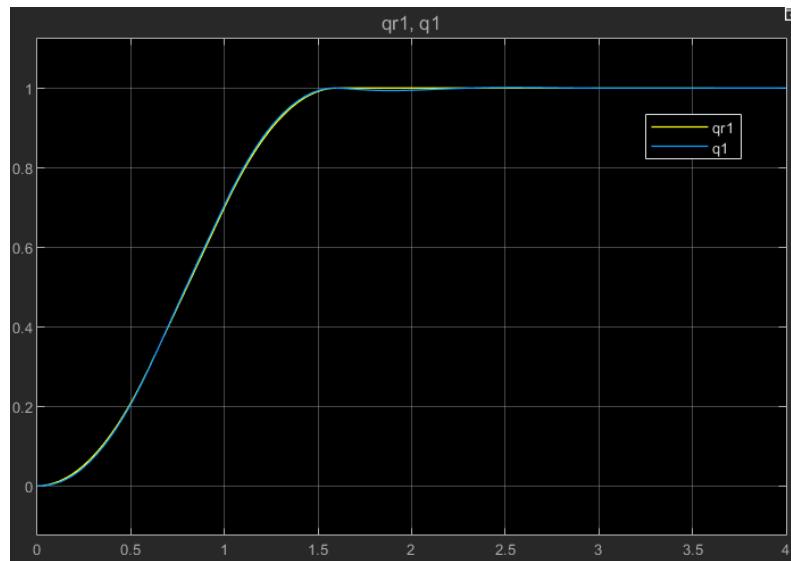
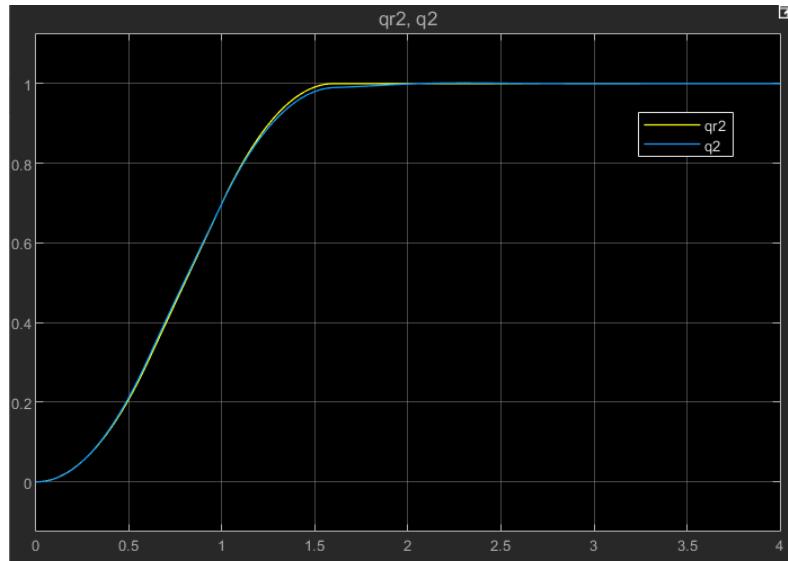


Figura 34: Implementazione Simulink coppia precalcolata - posizione e velocità

La simulazione è la seguente:





*Figura 35: Simulazione coppia precalcolata - posizione e velocità*

Si ottiene un ovvio miglioramento per entrambe le variabili di giunto. La capacità di inseguimento è migliorata in quanto non sono più presenti elevate oscillazioni e non è più presente il sovrasmorzamento per la prima variabile.

È ancora presente invece un sottosmorzamento, a  $t = 1.8s$  per la prima variabile e a  $t = 1.5s$  per la seconda variabile.

Quindi, è possibile ancora migliorare le prestazioni del controllo, valutando la compensazione a coppia precalcolata per lo schema di controllo completo delle tre retroazioni.

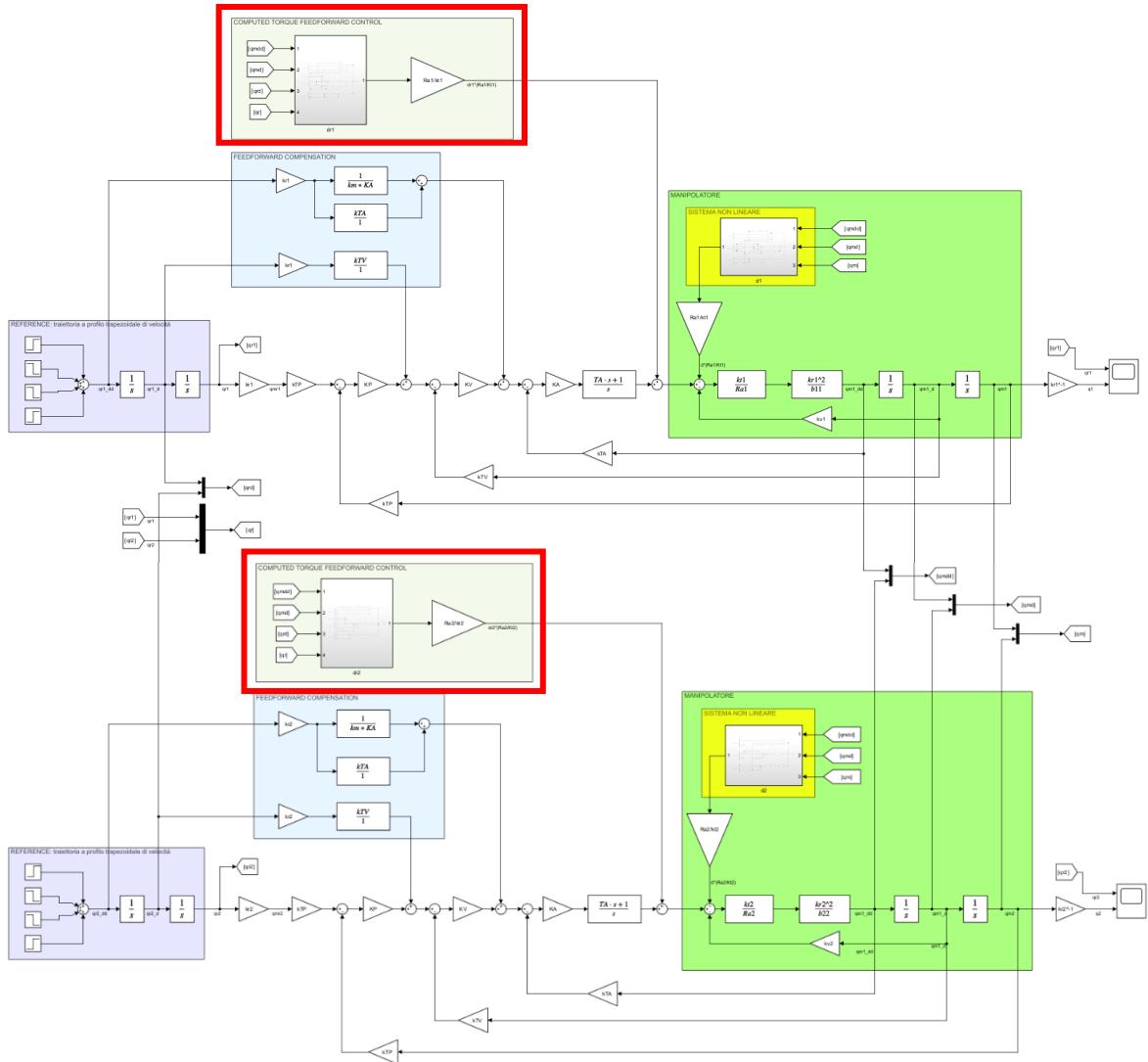
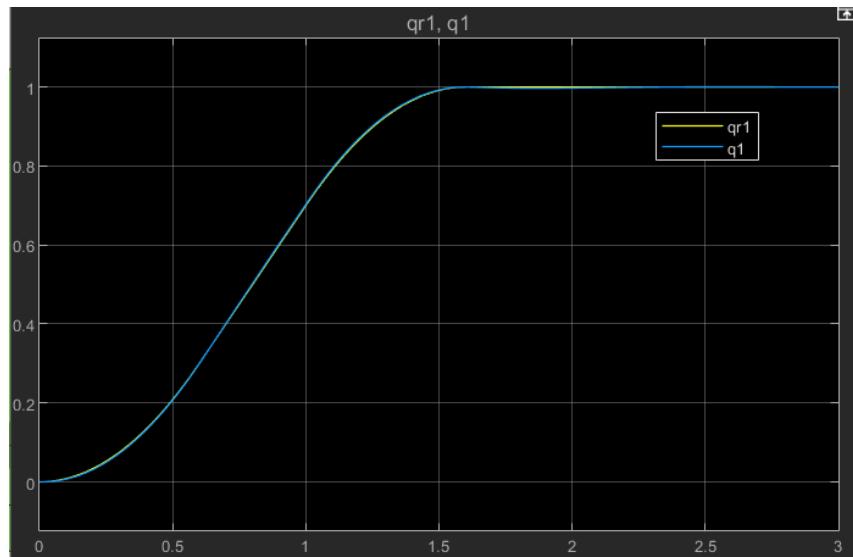


Figura 36: Implementazione Simulink coppia precalcolata - posizione, velocità e accelerazione

Infine, valutiamo la simulazione di quest'ultima implementazione.



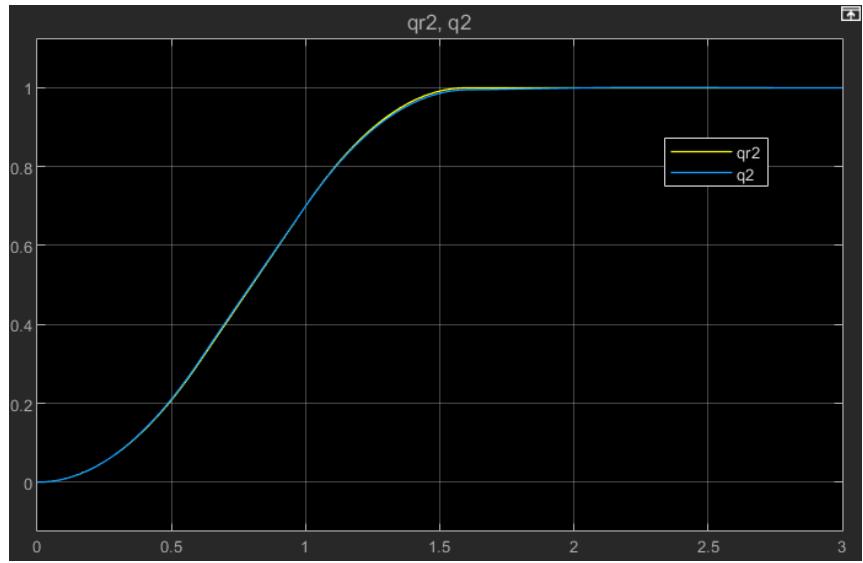


Figura 37: Simulazione coppia precalcolata – posizione, velocità e accelerazione

L'errore risulta invece:

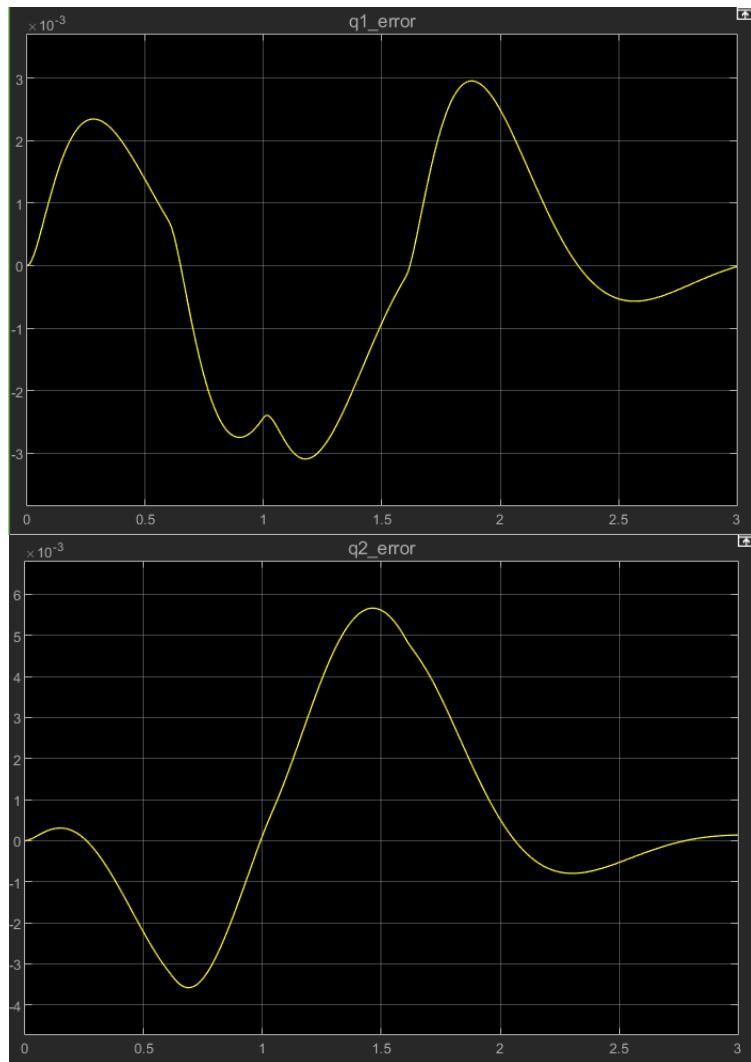


Figura 38: Errore coppia precalcolata – posizione, velocità e accelerazione

Si può ben notare che con l'utilizzo di tutte le retroazioni, posizione velocità e accelerazione, con la compensazione in avanti decentralizzata e infine la compensazione a coppia precalcolata, il controllo ha un'ottima performance in termini di inseguimento della traiettoria.

Infatti, si nota che entrambe le variabili di giunto inseguono con minimo errore la traiettoria a profilo trapezoidale di velocità, riuscendo ad assestarsi senza eccessivi sottosmorzamenti e sovrasmorzamenti.

È bene ricordare che comunque la presenza di azioni in avanti che utilizzano il modello dinamico del manipolatore possono gravare sul carico computazionale e quindi rallentare il controllo del sistema. Può risultare conveniente utilizzare un'azione in avanti parziale.

## 5.2 Controllo Centralizzato

### 5.2.1 Controllo PD con compensazione di gravità nello spazio dei giunti

Con il controllo PD con compensazione di gravità l'obiettivo è quello di trovare un vettore di controllo che assicuri l'asintotica stabilità globale della postura di equilibrio, imposta al manipolatore attraverso le variabili dei giunti desiderate  $q_d$ .

Date le equazioni di moto del manipolatore scritta nella seguente forma:

$$B(q) \ddot{q} + C(q, \dot{q}) \dot{q} + F\dot{q} + g(q) = u$$

Si assume un'azione di controllo costituita da un'azione di compensazione non lineare dei termini gravitazionali e un'azione lineare proporzionale-derivativa PD:

$$u = g(q) + K_P \tilde{q} - K_D \dot{q}$$

Dove le matrici  $K_P$  e  $K_D$  devono essere definite positive per assicurare la stabilità del sistema e  $\tilde{q}$  indica l'errore che si vuole minimizzare. L'errore  $\tilde{q}$  è definito come la differenza tra la variabile desiderata e la variabile stimata al tempo  $t$ :

$$\tilde{q} = q_d - q$$

Lo schema di controllo PD ai giunti con compensazione di gravità è il seguente:

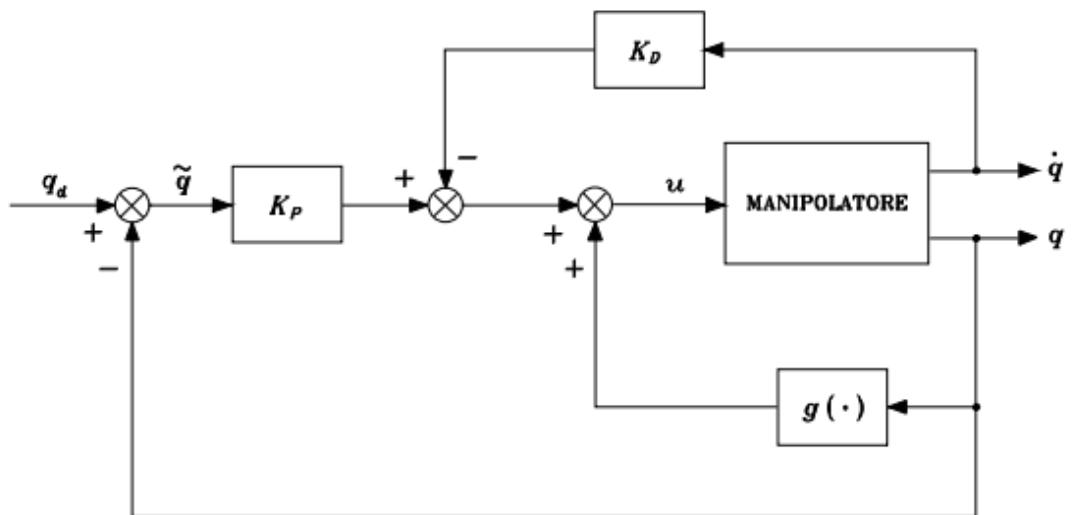


Figura 39: Schema di controllo PD ai giunti con compensazione di gravità

## Implementazione

Il manipolatore, come già detto, è caratterizzato da una postura iniziale:

$$q_0 = \begin{cases} \vartheta_1^0 = 30^\circ \approx 0.52 \text{ rad} \\ \vartheta_2^0 = 45^\circ \approx 0.79 \text{ rad} \end{cases}$$

Come postura di equilibrio da raggiungere, posizione di riferimento, è stata scelta la seguente:

$$q_d = \begin{cases} \vartheta_{d1} = 60^\circ \approx 1.05 \text{ rad} \\ \vartheta_{2d} = 25^\circ \approx 0.35 \text{ rad} \end{cases}$$

```
%> Condizioni iniziali
th1_0 = deg2rad(30); % rad
th1d_0 = 0; % rad/sec

th2_0 = deg2rad(45); % rad
th2d_0 = 0; % rad/sec

%> Postura costante
qd = [deg2rad(60); deg2rad(20)];
```

Figura 40: Condizioni iniziali e postura costante desiderata

Le costanti proporzionale e derivativa sono state scelte dopo un'accurata analisi *trial and error*, dove come obiettivo si è imposto la minimizzazione dell'errore  $\tilde{q}$ .

I valori sulla diagonale principale sono uguali in quanto sono stati considerati sia i link che i motori uguali.

Le matrici trovate sono le seguenti:

$$K_P = \begin{bmatrix} 3750 & 0 \\ 0 & 3750 \end{bmatrix} ; \quad K_D = \begin{bmatrix} 750 & 0 \\ 0 & 750 \end{bmatrix}$$

L'implementazione Simulink del controllo PD con compensazione di gravità è mostrata di seguito:

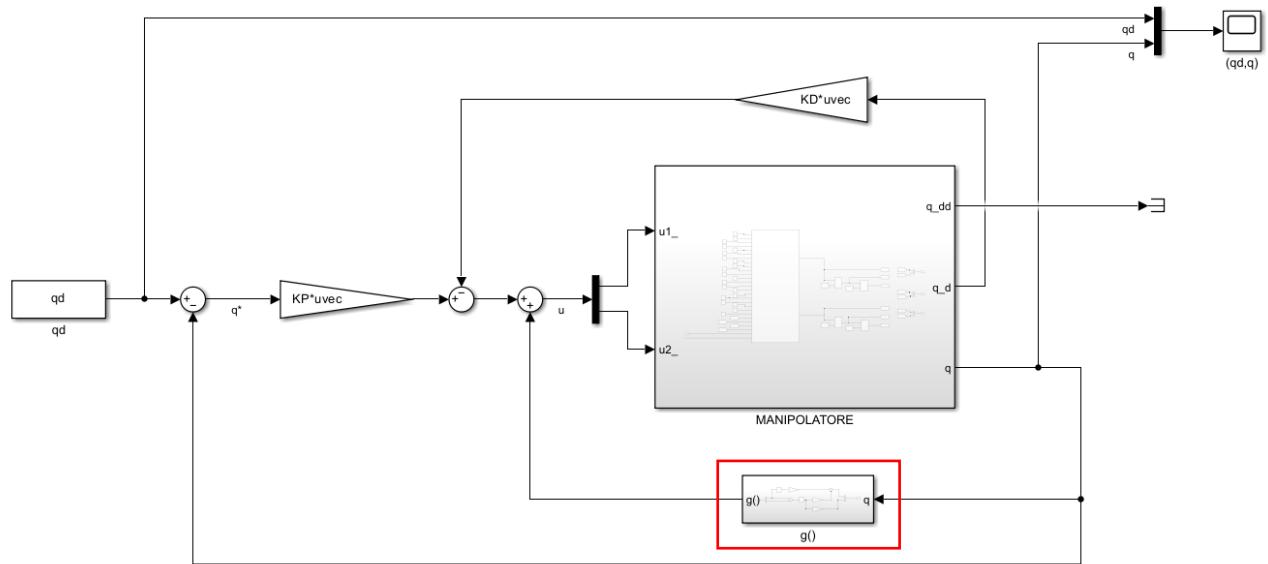


Figura 41: Implementazione Simulink del controllo PD con compensazione di gravità nello spazio dei giunti

Il blocco  $g()$  è stato implementato nel seguente modo:

$$g(q) = \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} = \begin{bmatrix} (m_1 d_1 + m_2 L_1 + m_{m2} L_1) * g c_1 + m_2 d_2 g c_{12} \\ m_2 d_2 g c_{12} \end{bmatrix} = \begin{bmatrix} g_{1\_1} c_1 + g_{1\_2} c_{12} \\ g_2 c_{12} \end{bmatrix}$$

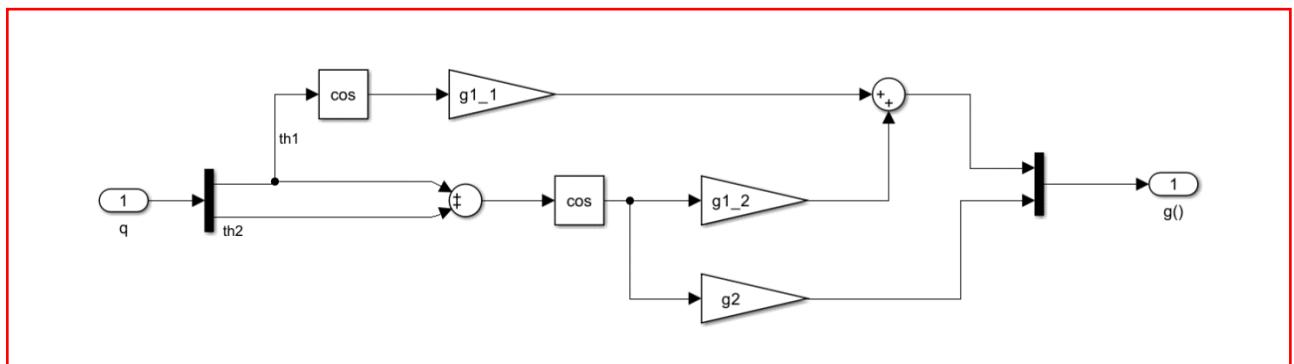


Figura 42: Implementazione compensazione di gravità

Sono state quindi confrontate le posizioni attuali dei due giunti  $q_1$  e  $q_2$  rispettivamente con quelle di riferimento  $qd_1$  e  $qd_2$  per valutare la risposta del sistema:

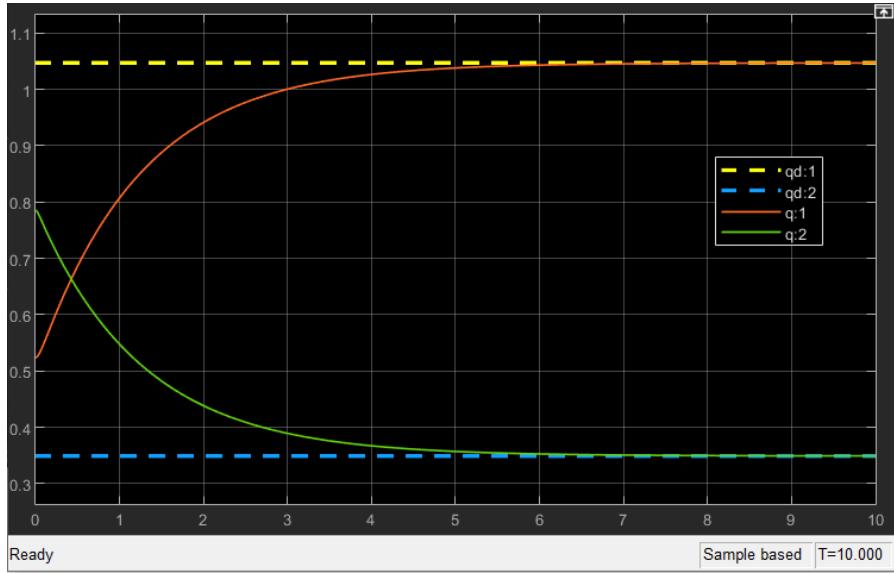


Figura 43: Simulazione controllo PD con compensazione di gravità  
nello spazio dei giunti

In particolare, il giunto 1 (in rosso) parte da una posizione iniziale  $\vartheta_1^0 = 30^\circ \approx 0.52$  rad al tempo  $t = 0$  e raggiunge la sua posizione desiderata di equilibrio, impostata a  $\vartheta_{d1} = 60^\circ \approx 1.05$  rad , circa al tempo  $t = 7.5$  s.

Stesso discorso per il secondo giunto (in verde), parte dalla posizione iniziale  $\vartheta_2^0 = 45^\circ \approx 0.79$  rad e raggiunge la sua posizione desiderata di equilibrio  $\vartheta_{d2} = 25^\circ \approx 0.35$  rad circa al tempo  $t = 7.5$  s.

## 5.2.2 Controllo a dinamica inversa nello spazio dei giunti

Con il controllo a dinamica inversa l'obiettivo è quello di inseguire una traiettoria desiderata.

Date le equazioni di moto del manipolatore scritta nella seguente forma:

$$B(q) \ddot{q} + n(q, \dot{q}) = u$$

Si assume un'azione di controllo costituita da un'azione lineare proporzionale-derivativa PD e un'azione di cancellazione della dinamica non lineare:

$$u = B(q) [K_P \tilde{q} + K_D \tilde{\dot{q}}] + B(q) \ddot{q}_d + n(q, \dot{q})$$

Dove le matrici  $K_P$  e  $K_D$  devono essere definite positive per assicurare la stabilità del sistema:

$$K_P = \begin{bmatrix} \omega_{n1}^2 & 0 & \dots & 0 \\ 0 & \omega_{n2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \omega_{nn}^2 \end{bmatrix}$$

$$K_D = \begin{bmatrix} 2\delta_1\omega_{n1} & 0 & \dots & 0 \\ 0 & 2\delta_2\omega_{n2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 2\delta_n\omega_{nn} \end{bmatrix}$$

Si ottiene un sistema disaccoppiato: la componente del riferimento influenza la sola variabile di giunto  $q_i$  con una relazione ingresso–uscita del secondo ordine caratterizzata da una pulsazione naturale  $\omega_{ni}$  e da un coefficiente di smorzamento  $\delta_i$ .

Lo schema di controllo a dinamica inversa è il seguente:

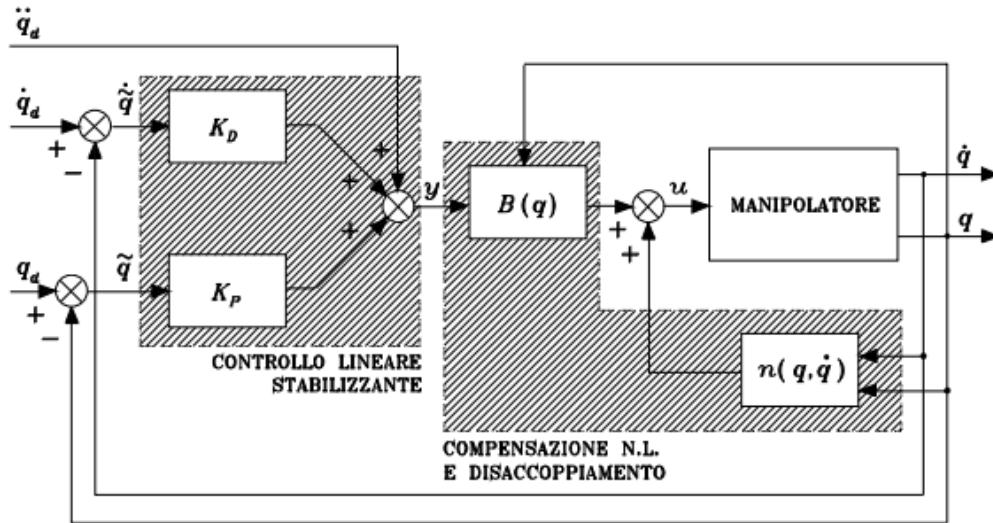


Figura 44: Schema di controllo a dinamica inversa ai giunti

### Implementazione

Le matrici  $K_P$  e  $K_D$  sono state definite considerando le pulsazioni naturali  $\omega_{n1}$  e  $\omega_{n2}$  uguali a  $5 \text{ rad/s}$  e i coefficienti di smorzamento  $\delta_1$  e  $\delta_2$  uguali a  $0.5$ .

Quindi, le costanti proporzionale e derivativa risultano:

$$K_P = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix} \quad ; \quad K_D = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

Come traiettoria di riferimento, per entrambi i giunti è stata utilizzata la traiettoria a profilo trapezoidale di velocità, descritta in precedenza nel paragrafo dedicato. L'accelerazione, velocità e la posizione di riferimento per i due giunti sono state indicate rispettivamente con le variabili  $qrdd$ ,  $qrd$  e  $qr$ .

L'implementazione Simulink del controllo a dinamica inversa è mostrata di seguito:

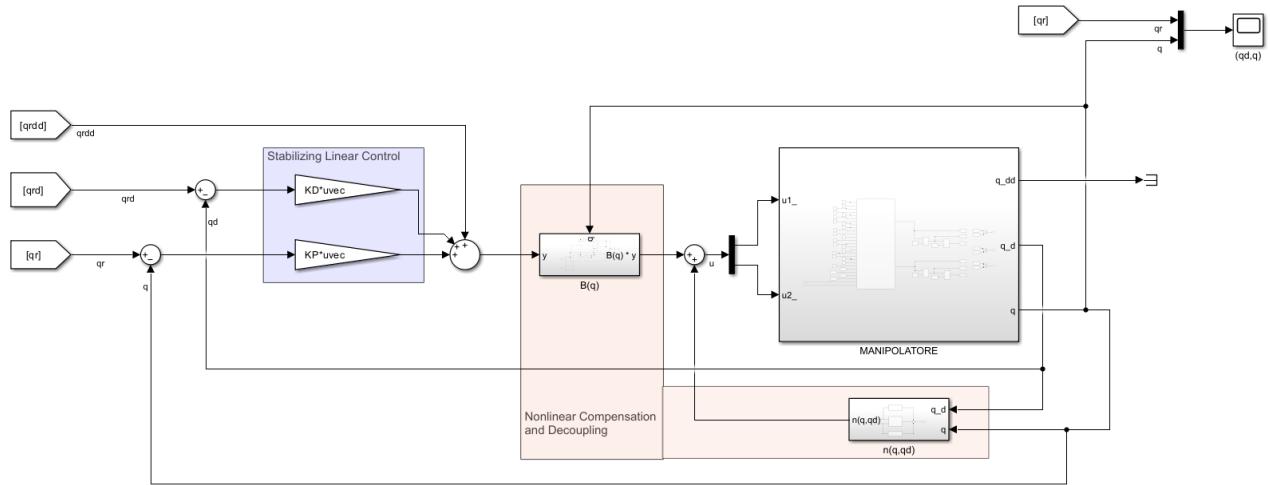


Figura 45: Implementazione Simulink del controllo a dinamica inversa nello spazio dei giunti

Il blocco  $B(q)$  è stato implementato nel seguente modo:

$$\begin{aligned}
 B(q) \ddot{\vartheta} &= \begin{bmatrix} B_{11}(q) & B_{12}(q) \\ B_{21}(q) & B_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{\vartheta}_1 \\ \ddot{\vartheta}_2 \end{bmatrix} = \begin{bmatrix} B_{11}(q)\ddot{\vartheta}_1 + B_{12}(q)\ddot{\vartheta}_2 \\ B_{21}(q)\ddot{\vartheta}_1 + B_{22}(q)\ddot{\vartheta}_2 \end{bmatrix} = \begin{bmatrix} (\mathbf{b}_{11} + \Delta\mathbf{b}_{11}\mathbf{c}_2)\ddot{\vartheta}_1 + (\mathbf{b}_{12} + \Delta\mathbf{b}_{12}\mathbf{c}_2)\ddot{\vartheta}_2 \\ (\mathbf{b}_{12} + \Delta\mathbf{b}_{12}\mathbf{c}_2)\ddot{\vartheta}_1 + \mathbf{b}_{22}\ddot{\vartheta}_2 \end{bmatrix} = \\
 &= \begin{bmatrix} [I_1 + m_1 d_1^2 + k_{r1}^2 I_{m1} + I_2 + m_2 (L_1^2 + d_2^2 + 2L_1 d_2 c_2) + I_{m2} + m_{m2} L_1^2] \ddot{\vartheta}_1 + [I_2 + m_2 (d_2^2 + L_1 d_2 c_2) + k_{r2} I_{m2}] \ddot{\vartheta}_2 \\ [I_2 + m_2 (d_2^2 + L_1 d_2 c_2) + k_{r2} I_{m2}] \ddot{\vartheta}_1 + [I_2 + m_2 d_2^2 + k_{r2} I_{m2}] \ddot{\vartheta}_2 \end{bmatrix}
 \end{aligned}$$

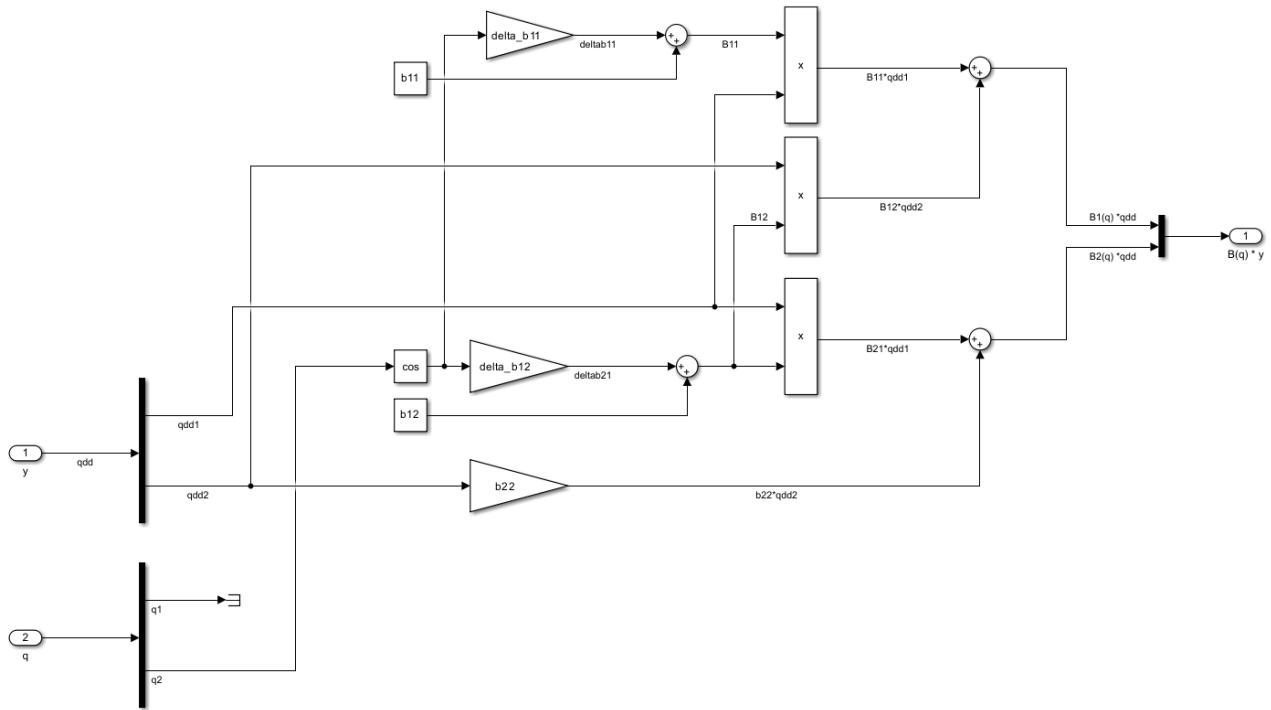


Figura 46: Implementazione in Simulink della matrice inerziale  $B(q)$

Il blocco  $n(q, qd)$  è stato implementato nel seguente modo:

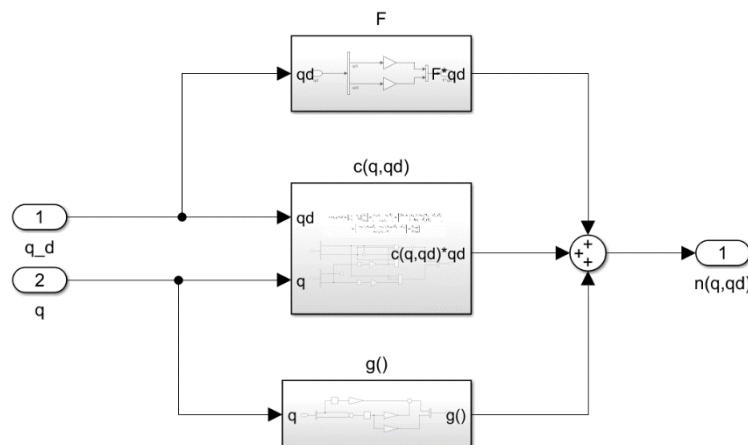


Figura 47: Implementazione in Simulink di  $n(q, q_d)$

Esso è costituito da tre Subsystem, rispettivamente  $F$ ,  $C(q, qd)$  e  $g()$ :

$$F\dot{\vartheta} = \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix} \begin{bmatrix} \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \end{bmatrix} = \begin{bmatrix} F_1 \dot{\vartheta}_1 \\ F_2 \dot{\vartheta}_2 \end{bmatrix} = \begin{bmatrix} \left( b_1 + \frac{k_{t1}k_{v1}k_{r1}^2}{R_{a1}} \right) \dot{\vartheta}_1 \\ \left( b_2 + \frac{k_{t2}k_{v2}k_{r2}^2}{R_{a2}} \right) \dot{\vartheta}_2 \end{bmatrix}$$

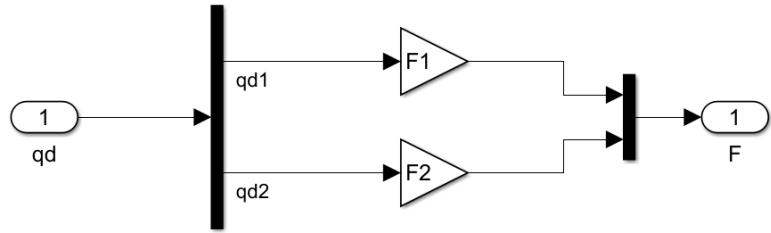


Figura 48: Implementazione in Simulink della matrice  $F$

$$\begin{aligned}
 C(q, qd) \dot{\vartheta} &= \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & 0 \end{bmatrix} \begin{bmatrix} \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \end{bmatrix} = \begin{bmatrix} c_{11}\dot{\vartheta}_1 + c_{12}\dot{\vartheta}_2 \\ c_{21}\dot{\vartheta}_1 \end{bmatrix} = \begin{bmatrix} (\mathbf{h}s_2\dot{\vartheta}_2)\dot{\vartheta}_1 + \mathbf{h}s_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)\dot{\vartheta}_2 \\ (-\mathbf{h}s_2\dot{\vartheta}_1)\dot{\vartheta}_1 \end{bmatrix} \\
 &= \begin{bmatrix} -m_2L_1d_2s_2\dot{\vartheta}_2 - m_2L_1d_2s_2(\dot{\vartheta}_1 + \dot{\vartheta}_2) \\ m_2L_1d_2s_2\dot{\vartheta}_1 \end{bmatrix} = \begin{bmatrix} C_{1,qd} \\ C_{2,qd} \end{bmatrix}
 \end{aligned}$$

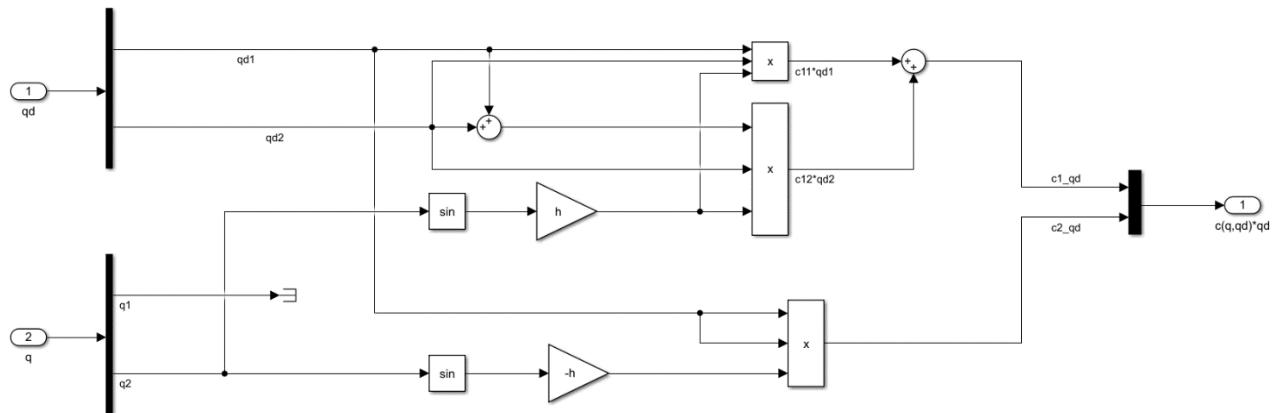
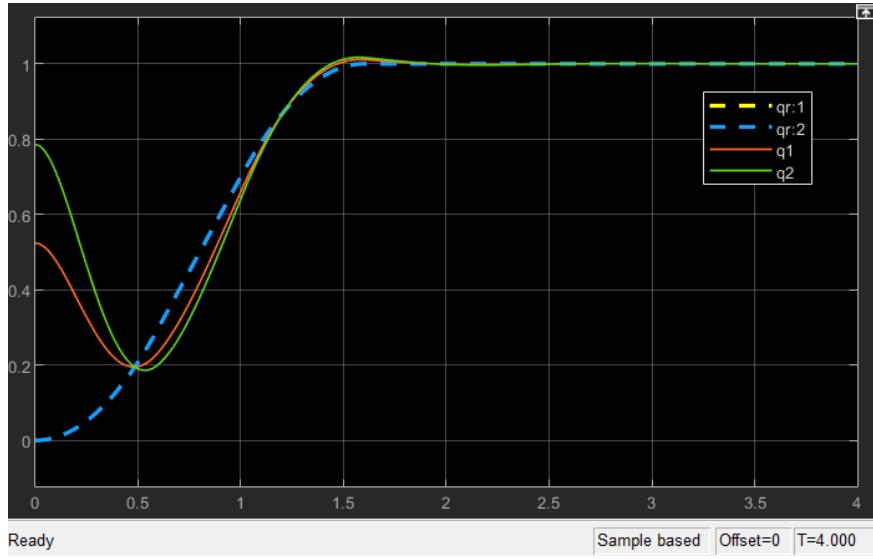


Figura 49: Implementazione in Simulink della matrice  $C(q, q_d)$

Il blocco  $g()$  è stato implementato come fatto nel precedente caso del controllo PD con compensazione di gravità.

Sono state confrontate le posizioni attuali dei due giunti ( $q_1$  e  $q_2$ ) rispettivamente con quelle di riferimento ( $qr_1$  e  $qr_2$ ).



*Figura 50: Simulazione controllo a dinamica inversa nello spazio dei giunti - 1*

I due giunti, rispettivamente in rosso e in verde, non riescono inizialmente a seguire la traiettoria ma presentano un andamento sottosmorzato a causa della loro posizione iniziale

$$q_0 = \begin{cases} \vartheta_1^0 = 30^\circ \approx 0.52 \text{ rad} \\ \vartheta_2^0 = 45^\circ \approx 0.79 \text{ rad} \end{cases}$$

Che è molto distante e non coincide al tempo  $t = 0s$  con la posizione inizializzata per la traiettoria desiderata

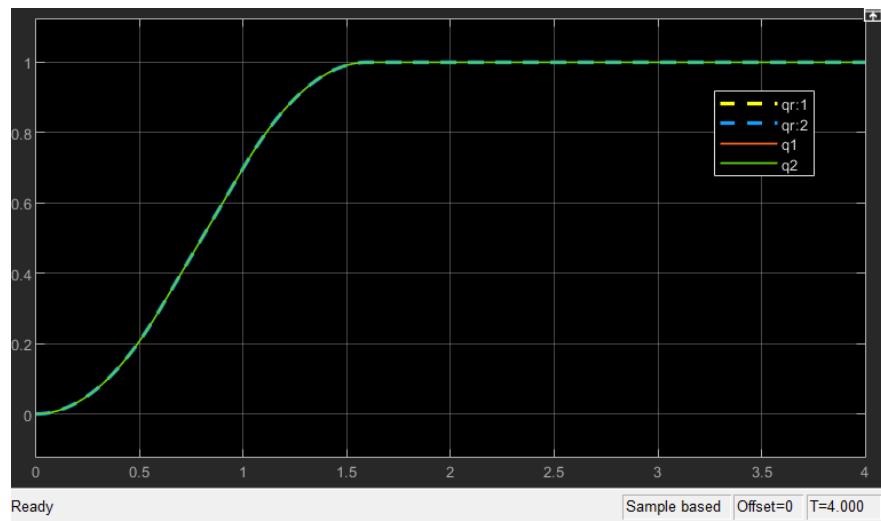
$$q_d^0 = \begin{cases} \vartheta_{d1}^0 = 0^\circ \approx 0 \text{ rad} \\ \vartheta_{d2}^0 = 0^\circ \approx 0 \text{ rad} \end{cases}$$

Si noti inoltre che al tempo  $t = 1.5s$  è presente un lieve sovrasmorzamento che caratterizza l'andamento dei due giunti.

È stata effettuata un'altra simulazione ma considerando le seguenti condizioni iniziali:

$$q_0 = \begin{cases} \vartheta_1^0 = 0^\circ \approx 0 \text{ rad} \\ \vartheta_2^0 = 0^\circ \approx 0 \text{ rad} \end{cases}$$

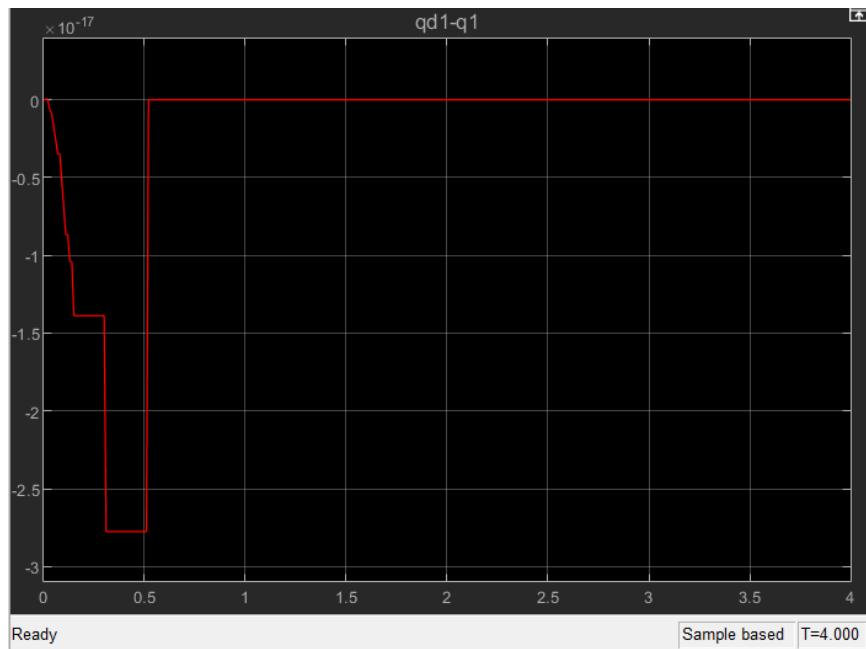
In modo che le condizioni iniziali coincidono con la posizione inizializzata per la traiettoria desiderata

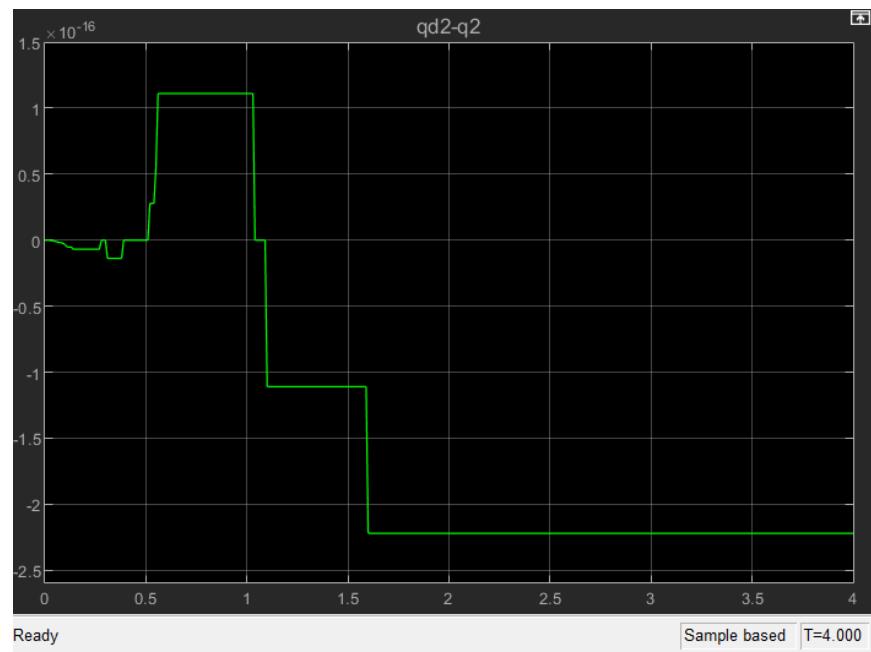


*Figura 51: Simulazione controllo a dinamica inversa nello spazio dei giunti - 2*

I due giunti in questo caso riescono a seguire perfettamente la traiettoria.

Si è valutato l'errore di posizione per entrambi i giunti  $\tilde{q} = q_d - q$





*Figura 52: Errore della simulazione del controllo a dinamica inversa  
nello spazio dei giunti - 2*

Gli errori risultano trascurabili rispettivamente di un ordine di grandezza pari a  $10^{-17}$  e  $10^{-16}$ .

### 5.2.3 Controllo robusto

Il controllo a dinamica inversa è caratterizzato da problemi di sensibilità e di robustezza. Il controllo robusto è una tecnica di controllo che contrasta gli effetti di compensazione imperfetta, presenti nel controllo a dinamica inversa sia per l'incertezza del modello che per le approssimazioni introdotte nel calcolo della dinamica inversa.

L'azione di controllo, rispetto al controllo a dinamica inversa, presenta un'azione  $\omega$  che conferisce robustezza allo schema di controllo:

$$u = \hat{B}(q) [K_P \tilde{q} + K_D \dot{\tilde{q}}] + \hat{B}(q) \ddot{q}_d + \hat{n}(q, \dot{q}) + \omega$$

Dove

$$\omega = \begin{cases} \frac{\rho}{\|z\|} z & \text{per } \|z\| \geq \epsilon \\ \frac{\rho}{\epsilon} z & \text{per } \|z\| < \epsilon \end{cases}$$

con

$$z = D^T Q \xi$$

in cui

$$\xi = \begin{bmatrix} \dot{\tilde{q}} \\ \ddot{\tilde{q}} \end{bmatrix}$$

$D$  è una matrice a blocchi di dimensione  $(2n \times n)$

$$D = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

Scelta una matrice simmetrica definita positiva  $P$  di dimensione  $(2n \times 2n)$ , la seguente equazione fornisce una soluzione  $Q$  unica, simmetrica e definita positiva di dimensione  $(2n \times 2n)$ :

$$\tilde{H}^T Q + Q \tilde{H} = -P$$

Dove

$$\tilde{H} = (H - DK)$$

In cui  $H$  è una matrice a blocchi di dimensione  $(2n \times 2n)$

$$H = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}$$

$$K = [K_P \quad K_D]$$

Dove le matrici  $K_P$  e  $K_D$  devono essere definite positive per assicurare la stabilità del sistema:

$$K_P = \begin{bmatrix} \omega_{n1}^2 & 0 & \dots & 0 \\ 0 & \omega_{n2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \omega_{nn}^2 \end{bmatrix}$$

$$K_D = \begin{bmatrix} 2\delta_1\omega_{n1} & 0 & \dots & 0 \\ 0 & 2\delta_2\omega_{n2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 2\delta_n\omega_{nn} \end{bmatrix}$$

Lo schema di controllo robusto ai giunti è il seguente:

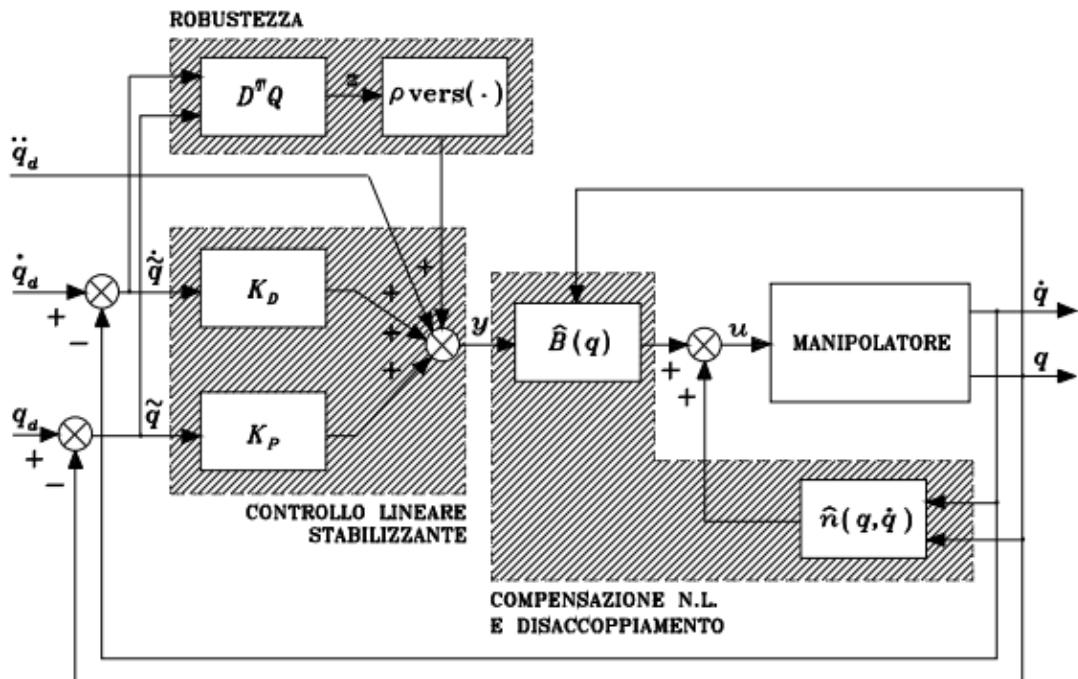


Figura 53: Schema di controllo robusto ai giunti

## Implementazione

Sono state considerate le ipotesi di inerzia costante  $\hat{B} = \bar{B}$  e compensazione di attrito meccanico e gravità  $\hat{n} = F_v \dot{q} + g$ , con i seguenti dati:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \rho = 70, \quad \epsilon = 0.001$$

$$K_P = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix}$$

$$K_D = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

$K_P$  e  $K_D$  sono state scelte come nel caso precedente.

Prima di implementare lo schema a blocchi sul *Simulink*, sul *Matlab* si è calcolato il termine  $z1 = D^T Q$  partendo dalla definizione delle matrici  $D, H, K$  e il calcolo della matrice  $\tilde{H}$ :

```
%Matrice D
D = zeros(4,2);
D(3,1) = 1;
D(4,2) = 1;

%Matrice H
H = zeros(4,4);
H(1,3) = 1;
H(2,4) = 1;

%Matrice K
K = [KP KD];

H_tilde = H - (D*K);
```

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad H = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$K = \begin{bmatrix} 25 & 0 & 5 & 0 \\ 0 & 25 & 0 & 5 \end{bmatrix}; \quad \tilde{H} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -25 & 0 & -5 & 0 \\ 0 & -25 & 0 & -5 \end{bmatrix}$$

Figura 54: Variabili controllo robusto Matlab

In seguito per trovare la matrice  $Q$  è stata risolta l'equazione  $\tilde{H}^T Q + Q \tilde{H} = -P$

$$Q = \begin{bmatrix} 2.7 & 0 & 0.020 & 0 \\ 0 & 2.7 & 0 & 0.020 \\ 0.020 & 0 & 0.104 & 0 \\ 0 & 0.020 & 0 & 0.104 \end{bmatrix}$$

```

%Definisco gli elementi della matrice Q come variabili qj
syms q1 q2 q3 q4
syms q5 q6 q7 q8
syms q9 q10 q11 q12
syms q13 q14 q15 q16
Qx = [q1 q2 q3 q4;
       q5 q6 q7 q8;
       q9 q10 q11 q12;
       q13 q14 q15 q16];

%Equazione da risolvere
equaz = (H_tilde')* Qx + Qx * H_tilde == -P;

%Creo un vettore che contiene gli elementi della matrice Q
vectq = [q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11 q12 q13 q14 q15 q16];

%Risolvo l'equazione rispetto alle variabili qj
tom = solve (equaz,vectq);

%Definisco la matrice Q sostituendo i qj con i valori trovati
vecttom = [tom.q1 tom.q2 tom.q3 tom.q4 tom.q5 tom.q6 tom.q7 tom.q8 tom.q9 tom.q10 tom.q11 tom.q12 tom.q13 tom.q14 tom.q15 tom.q16];
Q_sym = subs(Qx,vectq,vecttom);
Q = double(Q_sym);

```

Figura 55: Codice Matlab per determinazione matrice Q – controllo robusto

Infine, come anticipato è stato calcolato il termine  $z1 = D^T Q$

$$z1 = (D')^*(Q); \quad z1 = \begin{bmatrix} 0.020 & 0 & 0.104 & 0 \\ 0 & 0.020 & 0 & 0.104 \end{bmatrix}$$

Come traiettoria di riferimento, per entrambi i giunti è stata utilizzata la traiettoria a profilo trapezoidale di velocità, descritta in precedenza nel paragrafo dedicato. L'accelerazione, velocità e la posizione di riferimento per i due giunti sono state indicate rispettivamente con le variabili  $qrdd$ ,  $qrd$  e  $qr$ .

L'implementazione Simulink del controllo robusto è mostrata di seguito:

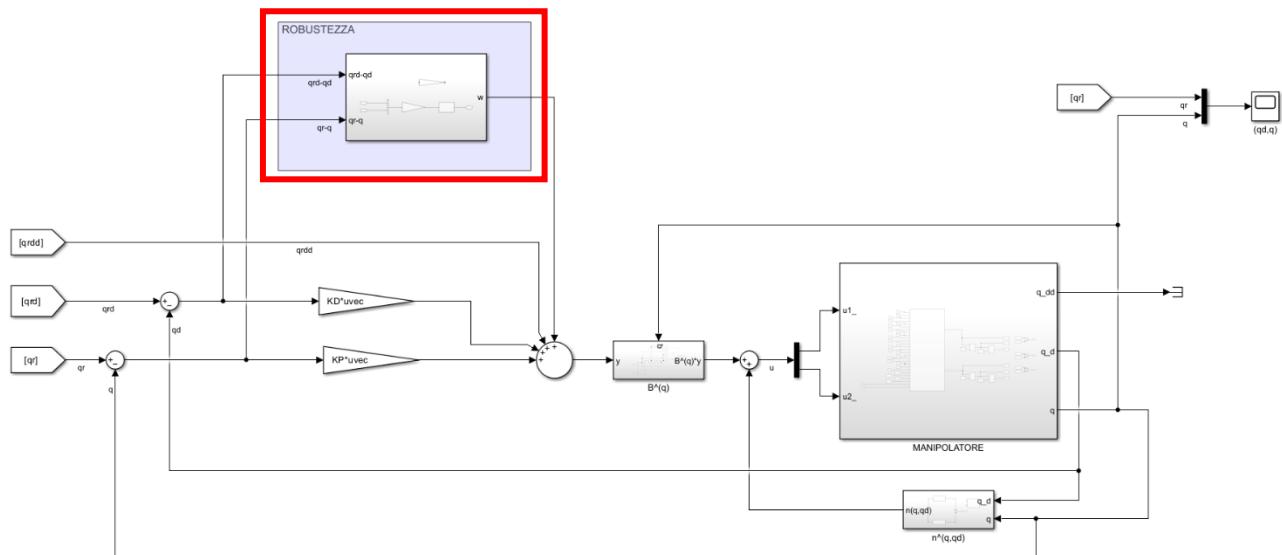


Figura 56: Implementazione Simulink del controllo robusto

Il blocco  $\hat{B}(q)$  è stato implementato nel seguente modo:

$$\begin{aligned}\hat{B}(q) \ddot{\vartheta} &= \begin{bmatrix} B_{11}(q) & 0 \\ 0 & B_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{\vartheta}_1 \\ \ddot{\vartheta}_2 \end{bmatrix} = \begin{bmatrix} B_{11}(q)\ddot{\vartheta}_1 \\ B_{22}(q)\ddot{\vartheta}_2 \end{bmatrix} = \begin{bmatrix} (\mathbf{b}_{11} + \Delta\mathbf{b}_{11}\mathbf{c}_2)\ddot{\vartheta}_1 \\ \mathbf{b}_{22}\ddot{\vartheta}_2 \end{bmatrix} = \\ &= \begin{bmatrix} [I_1 + m_1 d_1^2 + k_{r1}^2 I_{m1} + I_2 + m_2 (L_1^2 + d_2^2 + 2L_1 d_2 c_2) + I_{m2} + m_{m2} L_1^2]\ddot{\vartheta}_1 \\ (I_2 + m_2 d_2^2 + k_{r2} I_{m2})\ddot{\vartheta}_2 \end{bmatrix}\end{aligned}$$

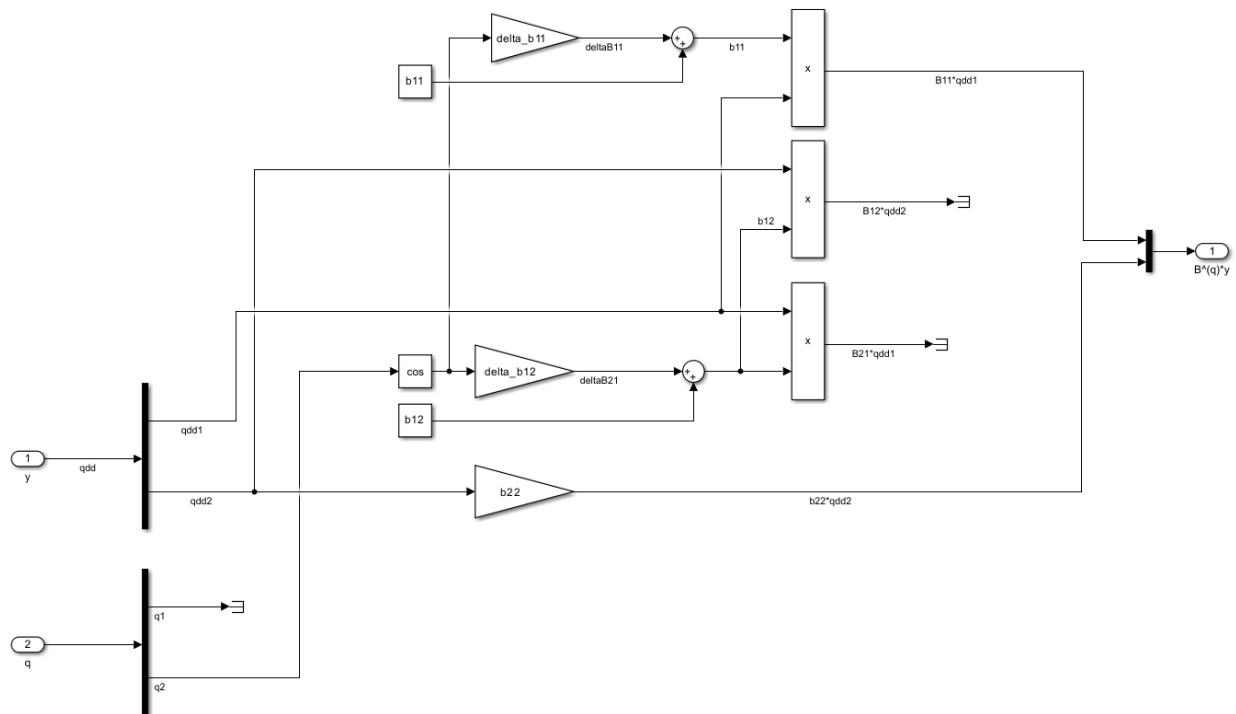


Figura 57: Implementazione del blocco  $\hat{B}(q)$

Il blocco  $\hat{n}(q, \dot{q})$  è stato implementato nel seguente modo:

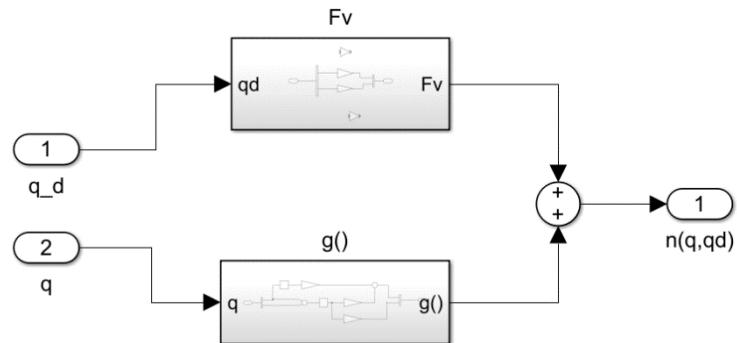


Figura 58: Implementazione di  $\hat{n}(q, \dot{q})$

Esso è costituito da due *Subsystem*, rispettivamente  $F_v$  e  $g()$ :

$$F_v \dot{\vartheta} = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix} \begin{bmatrix} \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \end{bmatrix} = \begin{bmatrix} b_1 \dot{\vartheta}_1 \\ b_2 \dot{\vartheta}_2 \end{bmatrix}$$

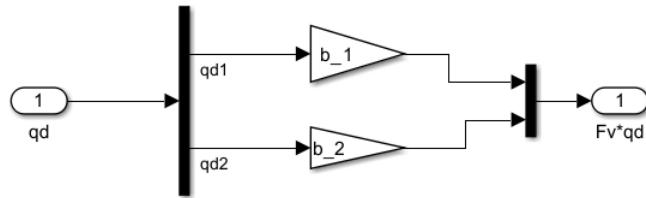


Figura 59: Implementazione della matrice dei coefficienti di smorzamento viscoso  $F_v$

Il blocco  $g()$  è stato implementato come già visto nei precedenti controlli.

Il blocco di robustezza è stato implementato come segue:

$$z = D^T Q \xi = z_1 \xi = z_1 \begin{bmatrix} \dot{\tilde{q}} \\ \tilde{q} \end{bmatrix} = z_1 \begin{bmatrix} \dot{q}_r - \dot{q} \\ q_r - q \end{bmatrix}$$

Dove  $\dot{q}_r$  e  $q_r$  sono rispettivamente la velocità e la posizione di riferimento per i due giunti;  $\dot{q}$  e  $q$  sono rispettivamente la velocità e la posizione attuale dei due giunti.

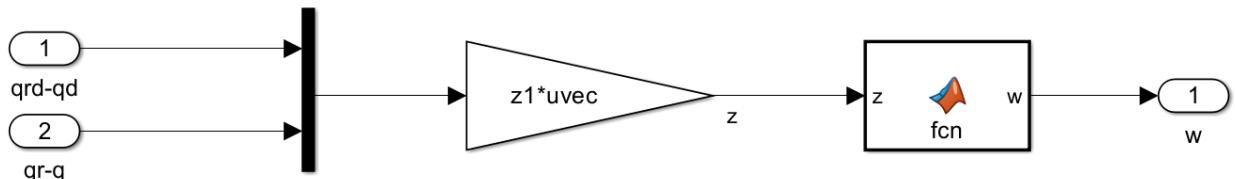


Figura 60: Implementazione della robustezza in Simulink

Caratterizzato dalla seguente funzione Matlab che implementa:

$$\omega = \begin{cases} \frac{\rho}{\|z\|} z & \text{per } \|z\| \geq \epsilon \\ \frac{\rho}{\epsilon} z & \text{per } \|z\| < \epsilon \end{cases}$$

```

1 function w = fcn(z)
2
3     ro = 70;
4     % PROVA 1:
5     % epsilon = 0.01;
6
7     % PROVA 2:
8     epsilon = 0.001;
9
10    if norm(z)>=epsilon
11        w = (z*ro)/norm(z);
12    else
13        w = (z*ro)/epsilon;
14    end

```

Figura 61: Funzione Matlab in Simulink che implementa  $\omega$

Sono state confrontate le posizioni attuali dei due giunti ( $q_1$  e  $q_2$ ) rispettivamente con quelle di riferimento ( $qr_1$  e  $qr_2$ ).

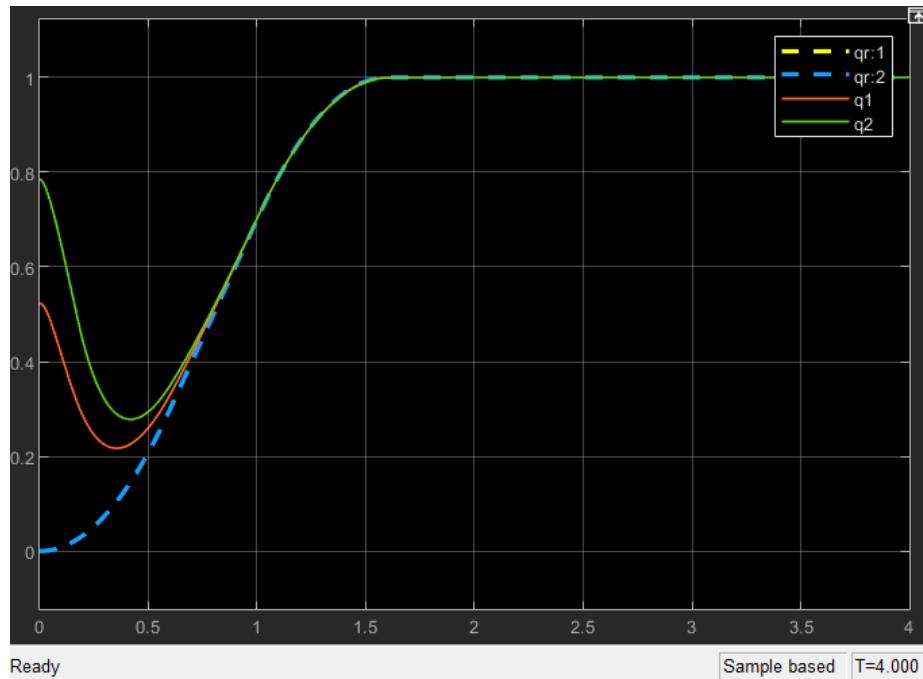


Figura 62: Simulazione controllo robusto nello spazio dei giunti - 1

I due giunti, rispettivamente in rosso e in verde, ovviamente non riescono inizialmente a seguire la traiettoria a causa della loro posizione iniziale:

$$q_0 = \begin{cases} \vartheta_1^0 = 30^\circ \approx 0.52 \text{ rad} \\ \vartheta_2^0 = 45^\circ \approx 0.79 \text{ rad} \end{cases}$$

Che non coincide, al tempo  $t = 0s$ , con la posizione inizializzata per la traiettoria desiderata

$$q_d^0 = \begin{cases} \vartheta_{d1}^0 = 0^\circ \approx 0 \text{ rad} \\ \vartheta_{d2}^0 = 0^\circ \approx 0 \text{ rad} \end{cases}$$

Confrontando questa simulazione con quella del controllo a dinamica inversa, si può dire che il controllo robusto si comporta meglio nella fase iniziale e anche nella fase intermedia in cui non presenta sovrasmorzamenti.

Osserviamo che considerando un  $\epsilon$  maggiore di quello considerato ( $\epsilon = 0.001$ ), ad esempio  $\epsilon = 0.01$ , otterremo un errore maggiore di un ordine di grandezza.

È stata effettuata un'altra simulazione ma considerando le seguenti condizioni iniziali:

$$q_0 = \begin{cases} \vartheta_1^0 = 0^\circ \approx 0 \text{ rad} \\ \vartheta_2^0 = 0^\circ \approx 0 \text{ rad} \end{cases}$$

In modo che le condizioni iniziali coincidono con la posizione inizializzata per la traiettoria desiderata.

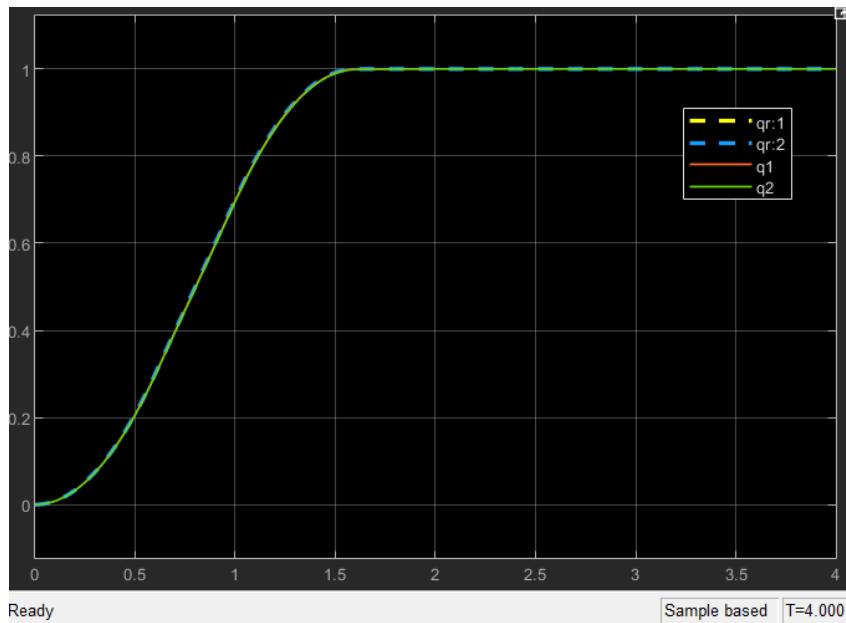
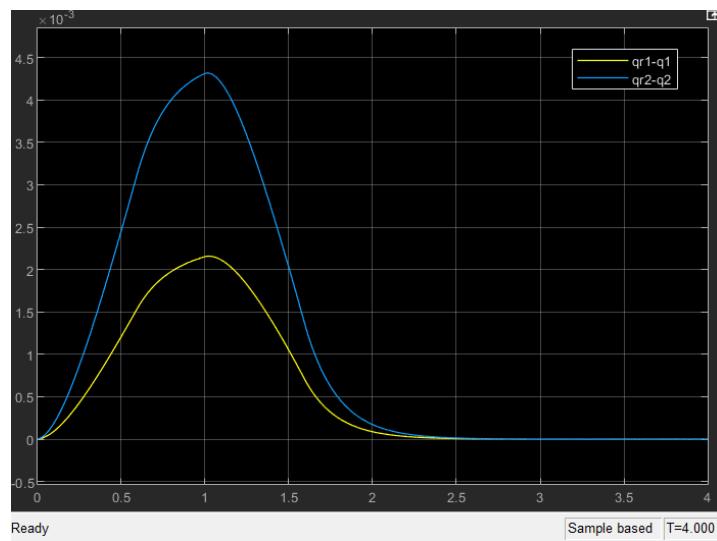


Figura 63: Simulazione controllo robusto nello spazio dei giunti - 2

I due giunti in questo caso riescono a seguire la traiettoria fin dall'inizio.

Si è valutato l'errore di posizione per entrambi i giunti  $\tilde{q} = q_d - q$



*Figura 64: Errore controllo robusto nello spazio dei giunti - 2*

Gli errori presentano, per entrambi i giunti, un picco al tempo  $t = 1s$  rispettivamente di circa  $2,2 \times 10^{-3}$  e  $4,3 \times 10^{-3}$ .

## 6. CONTROLLO NELLO SPAZIO OPERATIVO

### 6.1 Controllo PD con compensazione di velocità nello spazio operativo

Come visto nello spazio dei giunti, il controllo PD con compensazione di gravità ha come obiettivo quello di trovare un vettore di controllo che assicuri l'asintotica stabilità globale della postura di equilibrio, imposta al manipolatore attraverso la posizione nello spazio operativo desiderate  $x_d$ , e tale che l'errore  $\tilde{x} = x_d - x$  tenda asintoticamente a zero.

Date le equazioni di moto del manipolatore scritta nella seguente forma:

$$B(q) \ddot{q} + C(q, \dot{q}) \dot{q} + F\dot{q} + g(q) = u$$

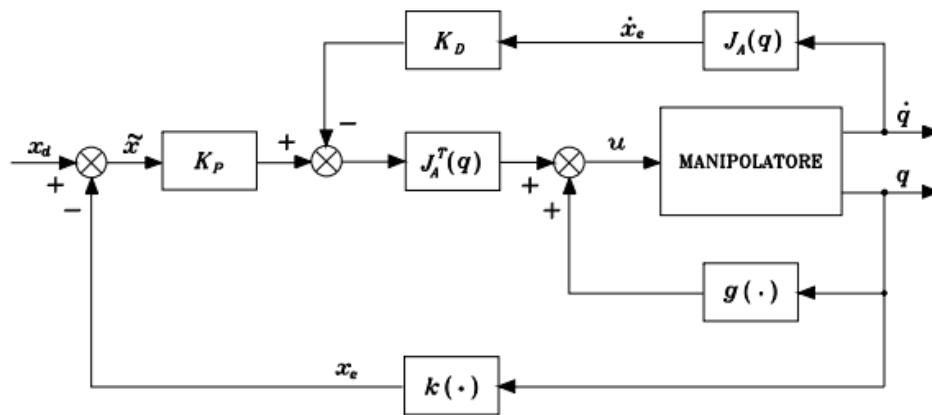
Si assume un'azione di controllo costituita da un'azione di compensazione non lineare dei termini gravitazionali nello spazio dei giunti e un'azione lineare proporzionale-derivativa PD nello spazio operativo:

$$u = g(q) + J_A^T(q)K_P \tilde{x} - J_A^T(q)K_D J_A^T(q)J_A(q)\dot{q}$$

L'ultimo termine viene aggiunto per aumentare lo smorzamento del sistema.

Le matrici  $K_P$  e  $K_D$  devono essere definite positive per assicurare la stabilità del sistema.

Lo schema di controllo PD ai giunti con compensazione di gravità è il seguente:



*Figura 65: Schema di controllo PD con compensazione di gravità nello spazio operativo*

## Implementazione

Per definire la postura nello spazio operativo, valutiamo lo spazio di lavoro che il manipolatore riesce a coprire, considerato che entrambi i bracci hanno una lunghezza di 1m :

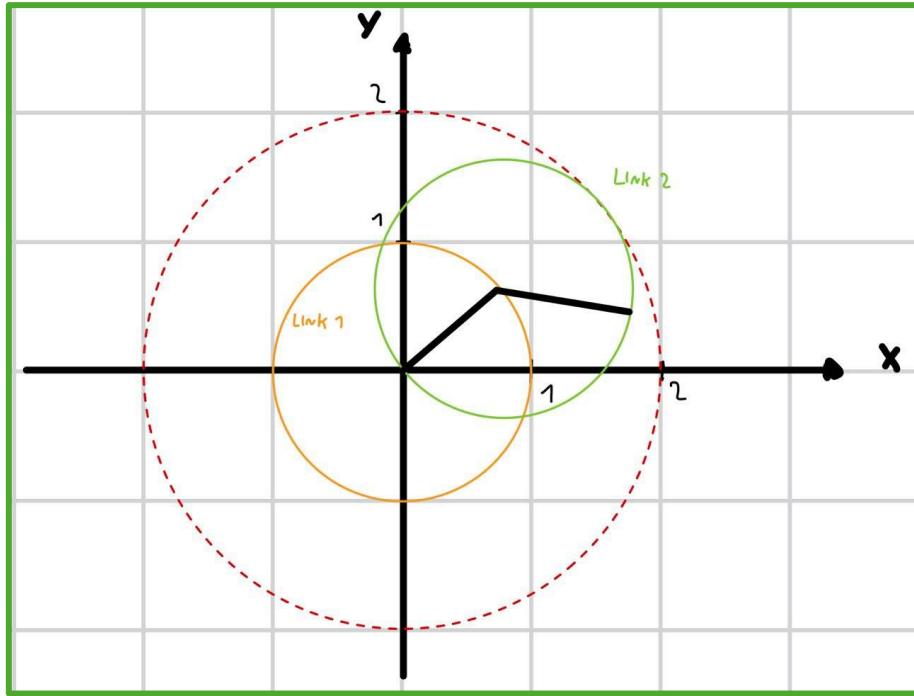


Figura 66: Rappresentazione grafica dello spazio raggiungibile dal manipolatore

Si nota che l'end-effector può operare entro una zona circolare di raggio 2m (circonferenza tratteggiata in rosso).

Il manipolatore è caratterizzato da una postura iniziale:

$$q_0 = \begin{cases} \vartheta_1^0 = 30^\circ \approx 0.52 \text{ rad} \\ \vartheta_2^0 = 45^\circ \approx 0.79 \text{ rad} \end{cases}$$

Per valutarla nello spazio operativo, ovvero valutare la posizione dell'end-effector:

$$P^{EE} = \begin{cases} x^{EE} = L_1 c_1 + L_2 c_{12} \\ y^{EE} = L_1 s_1 + L_2 s_{12} \end{cases}$$

Quindi, la postura iniziale dell'end-effector nello spazio operativo corrisponde a:

$$P_i^{EE} = \begin{cases} x_i^{EE} = 1.125 \text{ m} \\ x_i^{EE} = 1.466 \text{ m} \end{cases}$$

Come postura di equilibrio da raggiungere, posizione finale, è stata scelta la seguente:

$$q_d = \begin{cases} \vartheta_{d1} = 50^\circ \approx 0.8727 \text{ rad} \\ \vartheta_{2d} = 20^\circ \approx 0.3491 \text{ rad} \end{cases}$$

Che nello spazio operativo corrisponde a:

$$P_f^{EE} = \begin{cases} x_f^{EE} = 1.5 \text{ m} \\ y_f^{EE} = 0.5 \text{ m} \end{cases}$$

Nella successiva figura è stata schematizzata la posizione iniziale dell'end-effector (in blu) e la posizione finale che si desidera raggiungere (in verde).

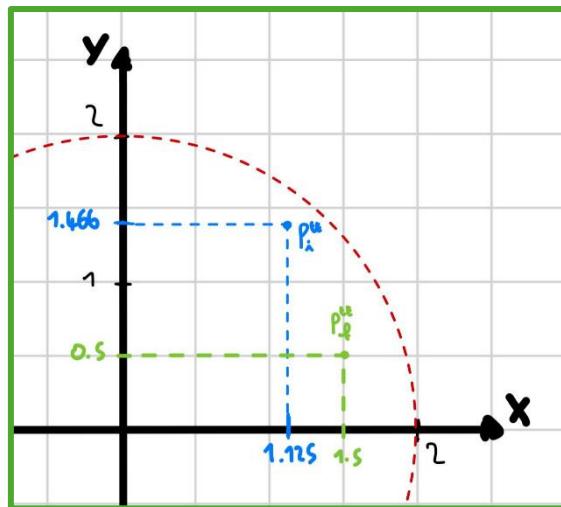


Figura 67: Rappresentazione posizione iniziale e finale dell'end-effector

Le matrici  $K_P$  e  $K_D$  sono state scelte come segue:

$$K_P = \begin{bmatrix} 16250 & 0 \\ 0 & 16250 \end{bmatrix} \quad ; \quad K_D = \begin{bmatrix} 3250 & 0 \\ 0 & 3250 \end{bmatrix}$$

Lo schema Simulink è il seguente:

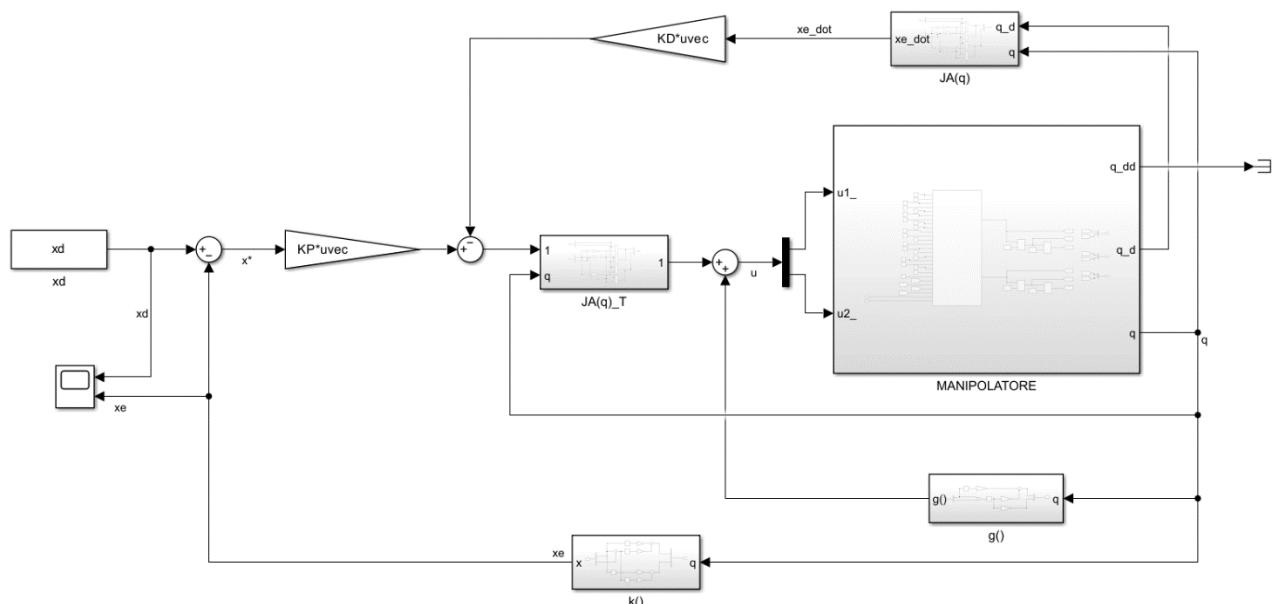


Figura 68: Implementazione Simulink del controllo PD con compensazione di gravità nello spazio operativo

Si possono notare, oltre ai blocchi già presentati nei capitoli precedenti, i blocchi  $k()$ ,  $J(q)$  e  $J(q)^T$ , che sono rispettivamente il blocco della cinematica diretta, lo Jacobiano analitico del manipolatore e lo Jacobiano analitico trasposto.

La funzione di cinematica diretta è stata implementata così:

$$\begin{cases} P_1^{EE} = p_x = L_1 c_1 + L_2 c_{12} \\ P_2^{EE} = p_y = L_1 s_1 + L_2 s_{12} \end{cases}$$

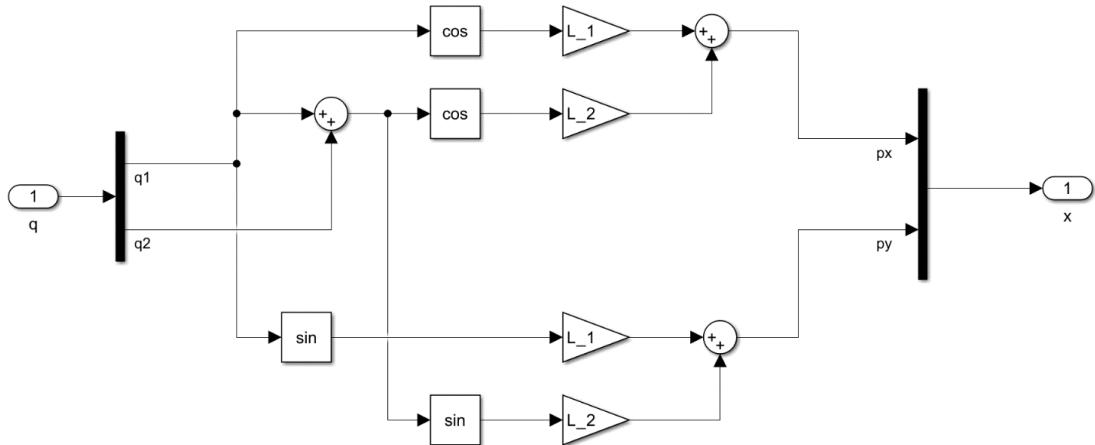


Figura 69: Implementazione Simulink della funzione di cinematica diretta  $k()$

Si nota che i due Jacobiani, analitico e geometrico, sono equivalenti in quanto il manipolatore planare può presentare rotazioni solo attorno all'asse  $z$ , quindi la rotazione dell'end-effector può avvenire solo intorno a un asse fisso nello spazio.

Lo Jacobiano analitico dell'end-effector è una matrice  $[3 \times 2]$  in quanto possono essere eliminate le tre righe nulle dello Jacobiano geometrico:

$$J_A(q) = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \\ 1 & 1 \end{bmatrix}$$

Lo Jacobiano trasposto  $[2 \times 3]$  è invece:

$$J_A(q)^T = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & L_1 c_1 + L_2 c_{12} & 1 \\ -L_2 s_{12} & L_2 c_{12} & 1 \end{bmatrix}$$

Il blocco dello Jacobiano e dello Jacobiano trasposto sono rispettivamente implementati come segue, interessandoci alla sola posizione dell'end-effector considereremo per  $J_A(q)$  solo le prime due righe e per  $J_A(q)^T$  solo le prime due colonne:

$$\dot{x}_e = J_A(q)\dot{q} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} J_1 \dot{q}_1 + J_2 \dot{q}_2 \\ J_3 \dot{q}_1 + J_4 \dot{q}_2 \end{bmatrix} = \begin{bmatrix} (-L_1 s_1 - L_2 s_{12})\dot{q}_1 + (-L_2 s_{12})\dot{q}_2 \\ (L_1 c_1 + L_2 c_{12})\dot{q}_1 + (L_2 c_{12})\dot{q}_2 \end{bmatrix}$$

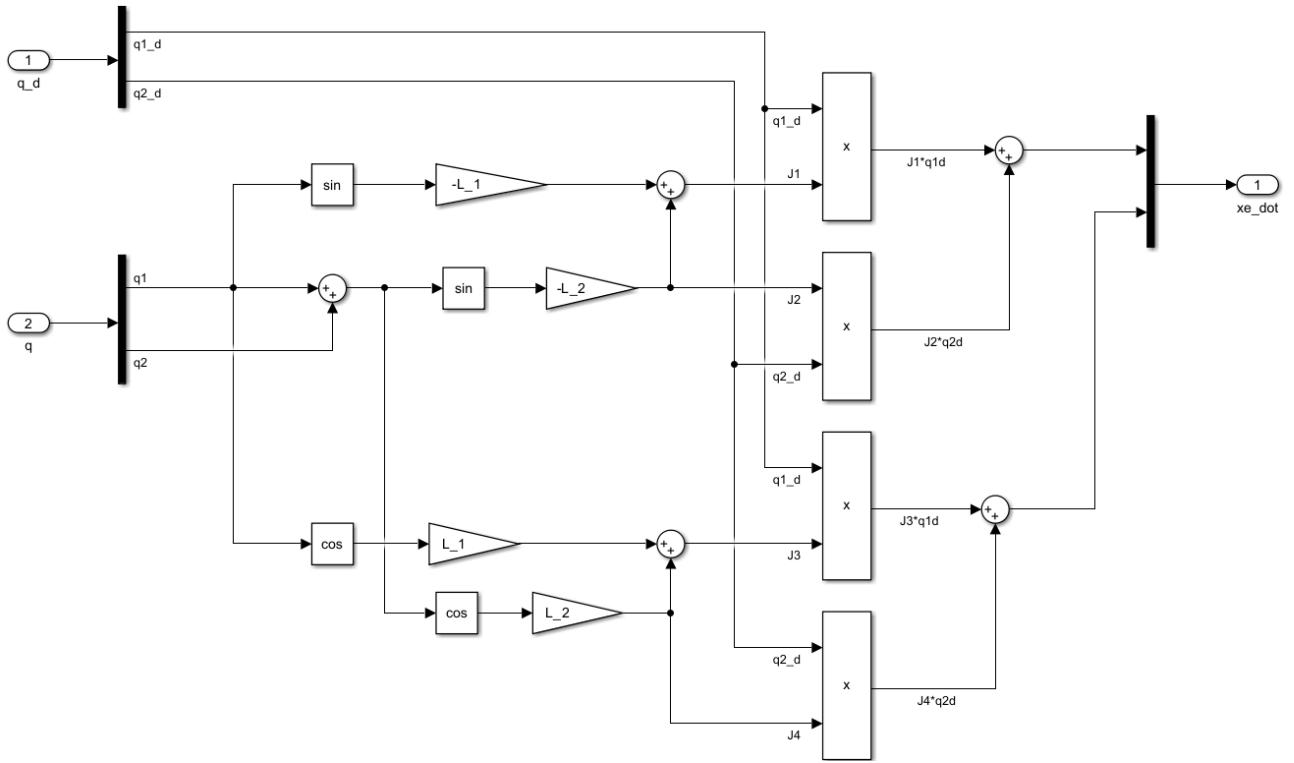


Figura 70: Implementazione Simulink Jacobiano analitico

(dove  $x_1$  e  $x_2$  sono gli input del Subsystem)

$$J_A(q)^T x = \begin{bmatrix} J_1 & J_3 \\ J_2 & J_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} J_1 x_1 + J_3 x_2 \\ J_2 x_1 + J_4 x_2 \end{bmatrix} = \begin{bmatrix} (-L_1 s_1 - L_2 s_{12}) x_1 + (L_1 c_1 + L_2 c_{12}) x_2 \\ -L_2 s_{12} x_1 + L_2 c_{12} x_2 \end{bmatrix}$$

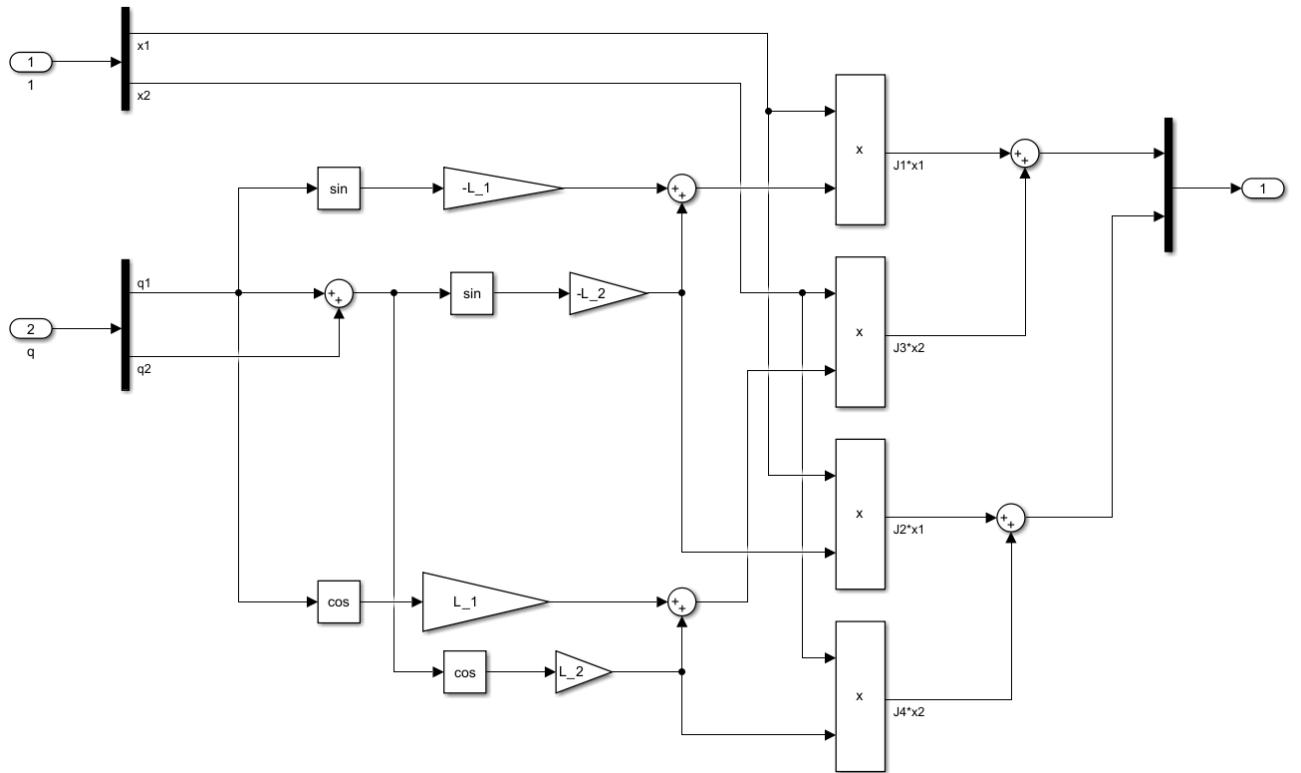


Figura 71: Implementazione Simulink Jacobiano analitico trasposto

La simulazione in  $t = 10\text{s}$  mostra come l'end-effector riesce agevolmente a raggiungere la coordinata  $y$  desiderata assestandosi dopo  $t = 5\text{s}$ . Invece per quanto riguarda la coordinata  $x$ , l'end-effector si assesta dopo  $t = 5\text{s}$  ed è caratterizzato da un andamento sovrasmorzato.

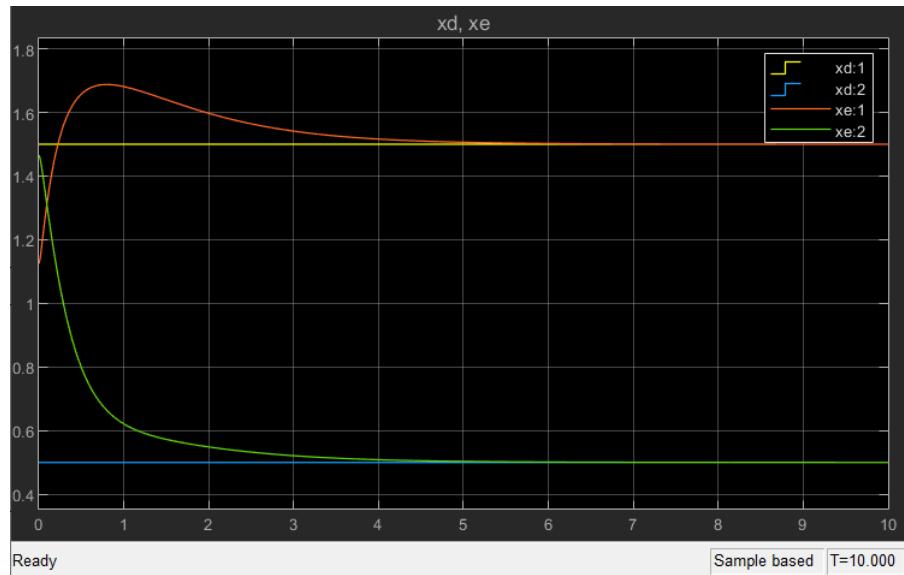


Figura 72: Simulazione controllo PD con compensazione di gravità nello spazio operativo

## 6.2 Controllo a dinamica inversa nello spazio operativo

Come per il caso nello spazio dei giunti, anche in questo caso ci si pone il problema di far inseguire una traiettoria al manipolatore, in questo caso definita nello spazio operativo.

Date le equazioni di moto del manipolatore scritta nella seguente forma:

$$B(q) \ddot{q} + n(q, \dot{q}) = u$$

Si assume l'azione di controllo costituita da un'azione lineare proporzionale-derivativa PD e un'azione di cancellazione della dinamica non lineare:

$$\begin{cases} u = B(q)y + n(q, \dot{q}) \\ y = J_A^{-1} [\ddot{x}_d + K_D \tilde{x} + K_P \tilde{x} - J_A(q, \dot{q})\dot{q}] \end{cases}$$

Dove le matrici  $K_P$  e  $K_D$  devono essere definite positive per assicurare la stabilità del sistema, in quanto determinano la velocità di convergenza dell'errore a zero:

$$K_P = \begin{bmatrix} \omega_{n1}^2 & 0 & \dots & 0 \\ 0 & \omega_{n2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \omega_{nn}^2 \end{bmatrix}$$

$$K_D = \begin{bmatrix} 2\delta_1\omega_{n1} & 0 & \dots & 0 \\ 0 & 2\delta_2\omega_{n2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 2\delta_n\omega_{nn} \end{bmatrix}$$

Lo schema di controllo a dinamica inversa nello spazio operativo è il seguente:

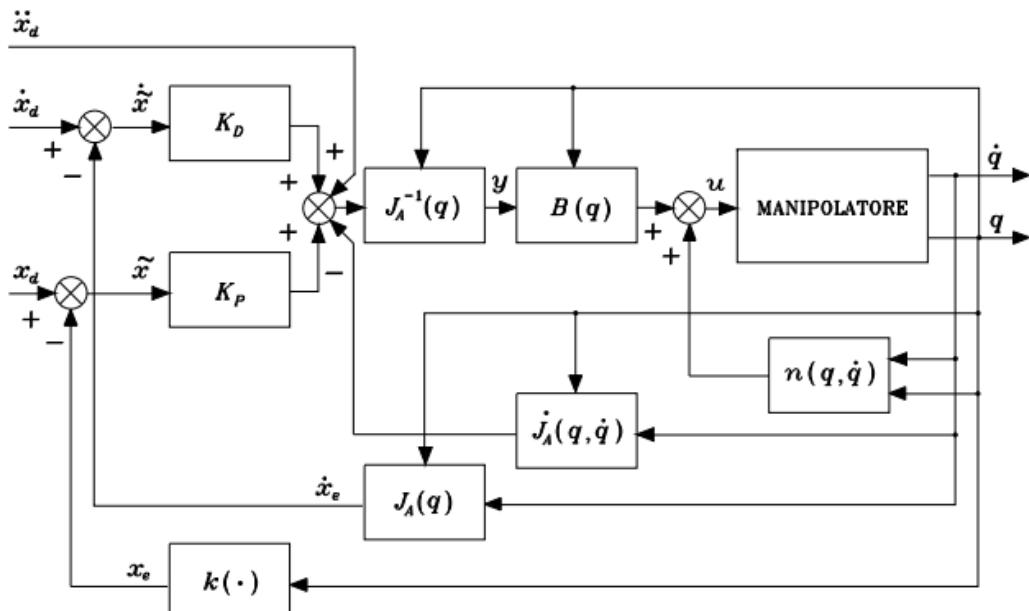


Figura 73: Schema di controllo a dinamica inversa nello spazio operativo

## Implementazione

Le matrici  $K_P$  e  $K_D$  sono state scelte uguali al caso di dinamica inversa nello spazio dei giunti, considerando le pulsazioni naturali  $\omega_{n1}$  e  $\omega_{n2}$  uguali a  $5 \text{ rad/s}$  e i coefficienti di smorzamento  $\delta_1$  e  $\delta_2$  uguali a 0.5.

Quindi, le costanti proporzionale e derivativa risultano:

$$K_P = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix} ; \quad K_D = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

L'implementazione Simulink del controllo a dinamica inversa è mostrata di seguito:

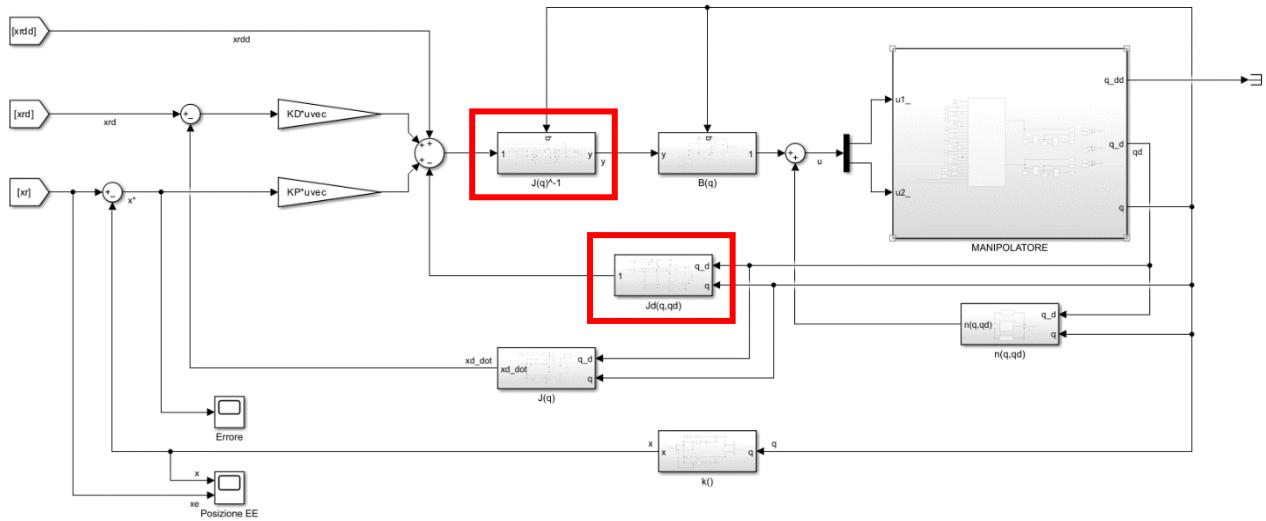


Figura 74: Implementazione Simulink a dinamica inversa nello spazio operativo

Oltre ai blocchi  $B(q)$ ,  $n(q, \dot{q})$ , illustrati nel paragrafo del controllo a dinamica inversa nello spazio dei giunti, il blocco della cinematica diretta  $k()$  e il blocco dello Jacobiano  $J_A(q)$ , visti nel paragrafo precedente, sono presenti due nuovi *Subsystem* (evidenziati in Figura [74]):

- il blocco dell'*inversa dello Jacobiano*  $J_A(q)^{-1}$
- il blocco della *derivata dello Jacobiano*  $J_A(q, \dot{q})$ .

L'espressione dell'inversa dello Jacobiano è:

$$J_A(q)^{-1} = \begin{bmatrix} -L_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)s_{12} & L_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)c_{12} \\ \frac{L_1L_2\dot{\vartheta}_1(\dot{\vartheta}_1 + \dot{\vartheta}_2)(c_1s_{12} + c_{12}s_1) + 2L_2^2(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2c_{12}s_{12}}{L_1\dot{\vartheta}_1s_1 + L_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)s_{12}} & \frac{L_1L_2\dot{\vartheta}_1(\dot{\vartheta}_1 + \dot{\vartheta}_2)(c_1s_{12} + c_{12}s_1) + 2L_2^2(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2c_{12}s_{12}}{-L_1\dot{\vartheta}_1c_1 - L_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)c_{12}} \\ \frac{L_1\dot{\vartheta}_1s_1 + L_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)s_{12}}{L_1L_2\dot{\vartheta}_1(\dot{\vartheta}_1 + \dot{\vartheta}_2)(c_1s_{12} + c_{12}s_1) + 2L_2^2(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2c_{12}s_{12}} & \frac{L_1L_2\dot{\vartheta}_1(\dot{\vartheta}_1 + \dot{\vartheta}_2)(c_1s_{12} + c_{12}s_1) + 2L_2^2(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2c_{12}s_{12}}{L_1L_2\dot{\vartheta}_1(\dot{\vartheta}_1 + \dot{\vartheta}_2)(c_1s_{12} + c_{12}s_1) + 2L_2^2(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2c_{12}s_{12}} \end{bmatrix}$$

In Simulink è implementato nel seguente modo (dove  $a_1$  e  $a_2$  sono gli input del subsystem):

$$J_A(q)^{-1} * \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \frac{1}{|J|} \begin{bmatrix} J_4 & -J_2 \\ -J_3 & J_1 \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \frac{J_4}{|J|} & -\frac{J_2}{|J|} \\ -\frac{J_3}{|J|} & \frac{J_1}{|J|} \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} J'_1 & J'_2 \\ J'_3 & J'_4 \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} J'_1 a_1 + J'_2 a_2 \\ J'_3 a_1 + J'_4 a_2 \end{bmatrix}$$

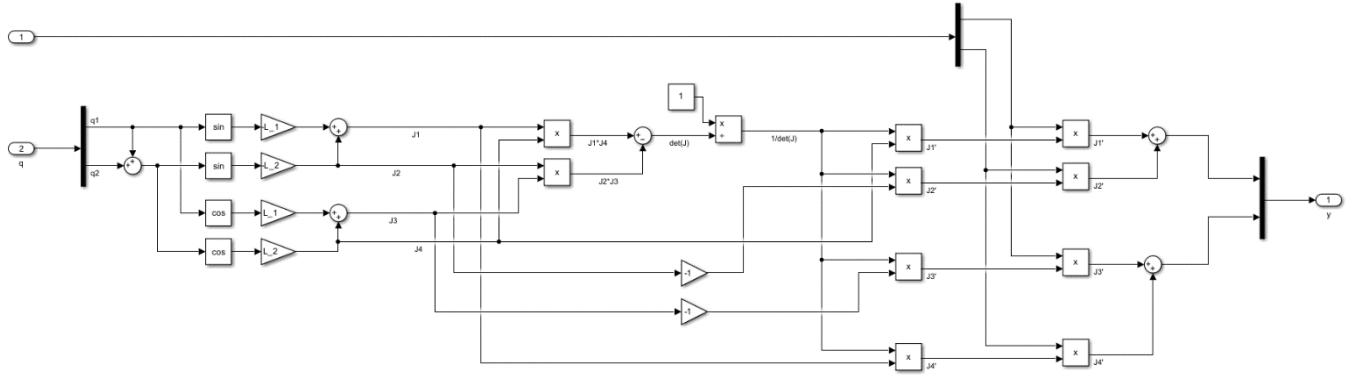


Figura 75: Implementazione Simulink inversa dello Jacobiano analitico

L'espressione della derivata dello Jacobiano è:

$$\dot{J}_A(q, \dot{q}) = \begin{bmatrix} -L_1 \dot{\vartheta}_1 c_1 - L_2 (\dot{\vartheta}_1 + \dot{\vartheta}_2) c_{12} & -L_2 (\dot{\vartheta}_1 + \dot{\vartheta}_2) c_{12} \\ -L_1 \dot{\vartheta}_1 s_1 - L_2 (\dot{\vartheta}_1 + \dot{\vartheta}_2) s_{12} & -L_2 (\dot{\vartheta}_1 + \dot{\vartheta}_2) s_{12} \end{bmatrix}$$

In Simulink è implementato così:

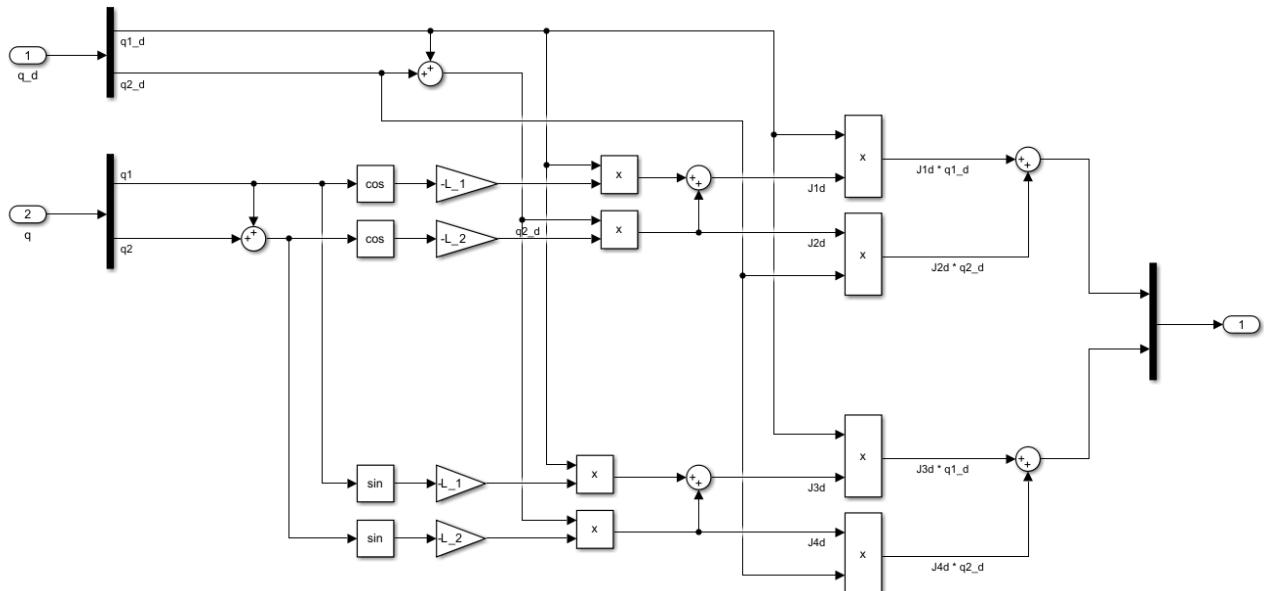


Figura 76: Implementazione Simulink della derivata dello Jacobiano analitico

$$J_A(q, \dot{q}) * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} J_1 \dot{q}_1 + J_2 \dot{q}_2 \\ J_3 \dot{q}_1 + J_4 \dot{q}_2 \end{bmatrix}$$

La simulazione è stata effettuata con la stessa traiettoria utilizzata nel controllo a dinamica inversa nello spazio dei giunti. Ovviamente, in questo caso, la traiettoria da seguire non indica i valori temporali che le variabili di giunto devono assumere, ma indica i valori della coordinata  $x$  e della coordinata  $y$  di riferimento nello spazio operativo.

Viene quindi confrontata la traiettoria di riferimento  $x_r$ , con la posizione effettiva del manipolatore, indicata con il vettore  $x_e$ . Il vettore  $x_e$  è l'uscita  $q$  del manipolatore moltiplicata per la funzione di cinematica diretta.

La posizione iniziale dell'end-effector è:

$$P_i^{EE} = \begin{cases} x_i^{EE} = 1.125 \text{ m} \\ y_i^{EE} = 1.466 \text{ m} \end{cases}$$

La simulazione è la seguente:

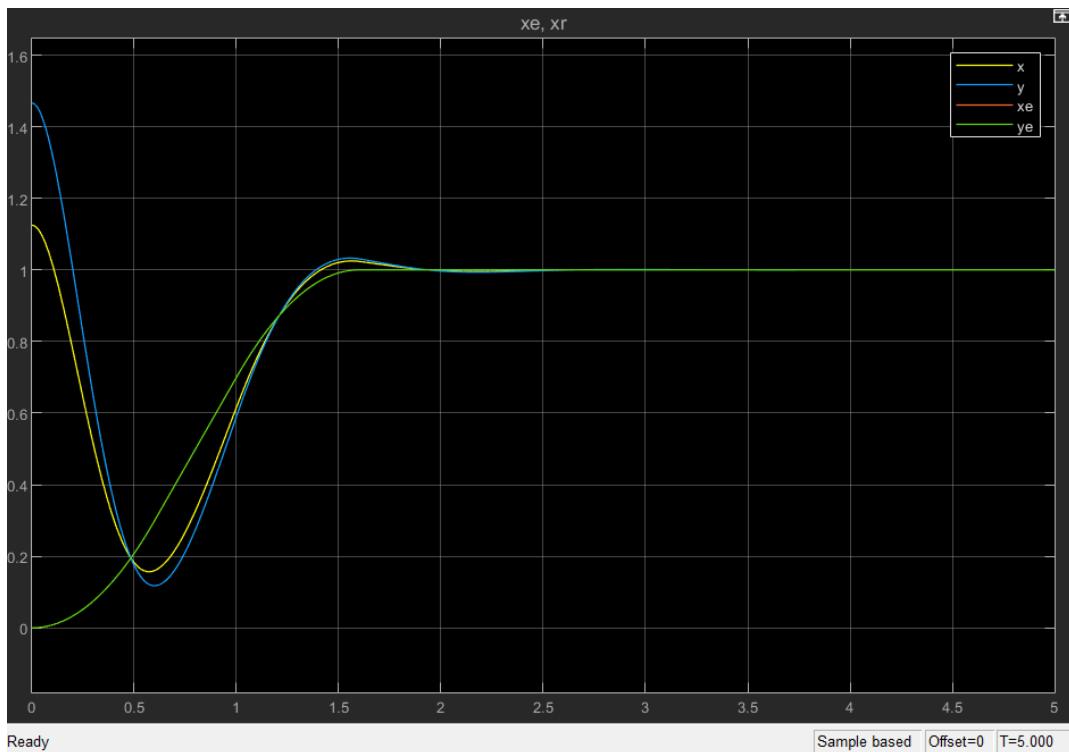


Figura 77: Simulazione controllo a dinamica inversa nello spazio operativo

Inizialmente, essendo la posizione iniziale del manipolatore diversa dalla posizione inizializzata nella traiettoria, si nota che le variabili  $x$  e  $y$  non riescono a seguire bene il riferimento.

Soltanto dopo 2 secondi riescono ad assestarsi, salvo un leggero sottosmorzamento nella fase iniziale e un sovrasmorzamento in quella intermedia, alla posizione finale richiesta.

Rivalutiamo l'algoritmo di controllo con il manipolatore che ha la stessa posizione iniziale della traiettoria, quindi per

$$P_i^{EE} = \begin{cases} x_i^{EE} = 0 \text{ m} \\ y_i^{EE} = 0 \text{ m} \end{cases}$$

Questo significa che il manipolatore ha come variabili di giunto  $\vartheta_1 = 0$  e  $\vartheta_2 = \pi$  ( $\vartheta_1$  potrebbe essere qualsiasi), quindi siamo in presenza di una posizione singolare del manipolatore.

La simulazione non si conclude con successo. Simulink restituisce un errore, suggerendo la possibile presenza di una singolarità nella soluzione.

## 7. CONCLUSIONI

I risultati peggiori si ottengono con il controllo indipendente ai giunti con retroazione di posizione. Infatti, l'assestamento lungo la traiettoria avviene solo dopo la sua fine, quindi con un elevato ritardo e con la presenza di un sovrasmorzamento.

Si nota come il controllo indipendente ai giunti con retroazione di posizione e di velocità presenta un errore elevato, causato dalla distanza della traiettoria effettivamente seguita dalle variabili di giunto da quella di riferimento. Le prestazioni sono quindi scadenti.

Si ottengono quasi gli stessi risultati con il controllo indipendente ai giunti con retroazione di posizione, velocità e accelerazione.

Si ottiene un miglioramento considerando gli algoritmi di controllo che utilizzano un'azione decentralizzata di compensazione, sia per il caso di sola retroazione di posizione, di posizione e velocità, di posizione velocità e accelerazione. Ovviamente, con quest'ultima configurazione si ottiene un risultato migliore, in quanto presenta un fattore di reiezione dei disturbi maggiore.

Il miglior risultato si ottiene con il controllo a coppia precalcolata, in quanto presenta un'azione di compensazione non lineare in avanti.

Possiamo inoltre confrontare queste prestazioni con le prestazioni del controllo PD ai giunti con compensazione di gravità. Il raggiungimento della postura finale del manipolatore avviene con prestazioni migliori rispetto ai controlli indipendenti ai giunti con le sole retroazioni, ma presenta un errore peggiore rispetto ai controlli indipendenti ai giunti in presenza dell'azione di compensazione in avanti e di coppia precalcolata.

Una migliore prestazione per quanto riguarda l'inseguimento della traiettoria si ottiene con il controllo a dinamica inversa, grazie a una linearizzazione globale ottenuta mediante una retroazione non lineare dello stato del sistema. Si deve comunque ricordare che per l'implementazione di questo schema di controllo è necessario il calcolo della matrice di inerzia  $B(q)$  e del vettore dei termini di Coriolis, centrifughi, gravitazionali e di smorzamento  $n(q, \dot{q})$ .

Per questo è interessante valutare le prestazioni del controllo robusto, che mira a contrastare le compensazioni imperfette del modello. Utilizzando un valore di soglia  $\epsilon$  piccolo si ottiene un errore di inseguimento contenuto, in quanto la traiettoria effettiva del manipolatore si avvicina a quella di riferimento. Aumentando la soglia, ovviamente aumenta l'errore, ma il controllo risulta meno aggressivo.

Infine, i controlli nello spazio operativo, quindi controllo PD con compensazione di gravità e controllo a dinamica inversa, hanno prestazioni praticamente simili se paragonati ai controlli omonimi nello spazio dei giunti.

Tuttavia, è indiscutibile che il controllo di un manipolatore nello spazio operativo risulti generalmente più complesso rispetto al controllo nello spazio dei giunti, poiché la

progettazione del controllore nello spazio operativo richiede comunque il calcolo dello Jacobiano del manipolatore.

Infatti, come si è potuto notare dalle simulazioni, la presenza di singolarità influenza lo Jacobiano e la sua derivata.

## 8. RIFERIMENTI

- [1] Sciavicco L and Siciliano B. Modelling and control of robot manipulators. 2nd ed. London: McGraw-Hill, 2000.
- [2] Orozco Manriquez EA. Modelling, identification and control of a two degrees-of-freedom robot manipulator (in Spanish). MSc Thesis, Instituto Politecnico Nacional-CITEDI, Mexico, 2009.
- [3] C.S.G. Lee, “Robot kinematics, dynamics and control,” *IEEE Computer* vol. 15, no. 12, pp. 62–80, 1982.
- [4] T.C.S. Hsia, T.A. Lasky, Z. Guo, “Robust independent joint controller design for industrial robot manipulators,” *IEEE Transactions on Industrial Electronics*, vol. 38, pp. 21–25, 1991.
- [5] P.K. Khosla, T. Kanade, “Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators,” *International Journal of Robotics Research*, vol. 7, no. 1, pp. 18–28, 1988.